

**IDENTIFIKASI UJARAN KEBENCIAN PADA FACEBOOK DENGAN
METODE *ENSEMBLE FEATURE* DAN *SUPPORT VECTOR MACHINE***

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Aditya Kresna Bayu Arda Putra

NIM: 135150200111028



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018

PENGESAHAN

IDENTIFIKASI UJARAN KEBENCIAN PADA FACEBOOK DENGAN METODE
ENSEMBLE FEATURE DAN SUPPORT VECTOR MACHINE

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan

Memperoleh gelar sarjana computer

Disusun Oleh:

Aditya Kresna Bayu A.P.

NIM: 135150200111028

Skripsi telah diuji dan dinyatakan lulus pada

27 Juli 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

M. Ali Fauzi, S.Kom, M.Kom

NIK. 201502 890101 1 001

Budi Darma Setiawan, S.Kom, M.Cs

NIP. 198410 15201404 1 002

Mengetahui

Ketua Jurusan Teknik Informatika



Tri Astoto Kumiawan, S.T, M.T, Ph.D

NIP. 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 03 Agustus 2018



Aditya Kresna Bayu A.P.

NIM: 135150200111028

KATA PENGANTAR

Puji syukur kami panjatkan kehadiran Tuhan Yang Maha Esa, karena berkat rahmat dan hidayah-Nya, penulis mampu menyelesaikan laporan Skripsi yang berjudul “Identifikasi Ujaran Kebencian pada *Facebook* dengan Metode *Ensemble Feature* dan *Support Vector Machine*”.

Skripsi adalah salah satu mata kuliah yang wajib ditempuh sebagai syarat kelulusan di Fakultas Ilmu Komputer Universitas Brawijaya. Laporan ini disusun sebagai syarat kelulusan dalam menempuh studi di Universitas Brawijaya.

Dalam penyusunan laporan ini penulis mendapatkan banyak bantuan dari berbagai pihak baik secara moril maupun materil. Dalam kesempatan ini penulis ingin mengucapkan terimakasih yang sebesar-besarnya kepada:

1. Bapak Mochammad Ali Fauzi, S.Kom, M.Kom, selaku Dosen Pembimbing 1 dan Bapak Budi Darma Setiawan, S.Kom, M.Cs, selaku Dosen Pembimbing 2 yang telah membimbing, memberi saran serta motivasi kepada penulis selama penyusunan skripsi ini.
2. Seluruh dosen dan civitas akademika Fakultas Ilmu Komputer yang telah memberikan wawasan, bantuan dan dukungan yang sangat bermanfaat bagi penulis.
3. Kedua orang tua penulis, atas segala dukungan yang diberikan kepada penulis, terutama Ibu saya.
4. Teman-teman dari EROrganization yang telah banyak membantu kelancaran pelaksanaan dan penyusunan laporan Skripsi ini.

Kami menyadari bahwa dalam penyusunan laporan Skripsi ini masih banyak kekurangan baik dari segi susunan maupun isi. Oleh karena itu, saya sebagai penulis dengan senang hati akan menerima kritik dan saran yang membangun sebagai pedoman perbaikan kedepannya.

Malang, 03 Agustus 2018

Penyusun

kresna.9962@gmail.com

ABSTRAK

Aditya Kresna Bayu A.P., Identifikasi Ujaran Kebencian Pada Facebook dengan Metode Ensemble Feature dan Support Vector Machine

Pembimbing : M. Ali Fauzi, S.Kom., M.Kom. dan Budi Darma Setiawan, S.Kom., M.Sc.

Pada awalnya, media sosial dipergunakan untuk bersosialisasi dan berinteraksi dengan orang lain. Salah satu media sosial yang banyak digunakan orang-orang untuk bersosialisasi adalah Facebook, dengan jumlah pengguna mencapai lebih dari ratusan juta orang yang tersebar di seluruh dunia. Akhir-akhir ini, pada media sosial Facebook, sering kali terdapat tulisan yang berisikan ujaran kebencian yang dibagikan secara meluas. Tentunya dibutuhkan bantuan ahli Bahasa untuk mengidentifikasi ujaran kebencian pada media sosial karena belum ada sebuah sistem otomatis untuk mengidentifikasi ujaran kebencian. Sistem yang ada dalam penelitian ini dibuat dengan menggunakan metode *Ensemble Feature* dan *Support Vector Machine* (SVM). Metode *Ensemble Feature* dapat dipergunakan untuk menggabungkan beberapa fitur dari tiap tulisan tersebut untuk mempermudah proses identifikasi ujaran kebencian. Metode *Support Vector Machine* (SVM) kemudian digunakan untuk melakukan proses identifikasi terhadap sebuah tulisan berdasarkan data-data fitur yang dihasilkan oleh metode *Ensemble Feature*. Berdasarkan hasil dari proses pengujian, diperoleh akurasi sistem sebesar 70% sehingga dapat disimpulkan bahwa metode *Ensemble Feature* dan *Support Vector Machine* (SVM) baik untuk digunakan dalam proses identifikasi ujaran kebencian pada media sosial Facebook.

Kata kunci: *Ensemble Feature, Support Vector Machine, Ujaran Kebencian, Facebook*

ABSTRACT

Aditya Kresna Bayu A.P., *Hatespeech Identification on Facebook Using Ensemble Features and Support Vector Machine*

Supervisors : M. Ali Fauzi, S.Kom., M.Kom. dan Budi Darma Setiawan, S.Kom., M.Sc.

In the beginning, social media is used for socializing and interacting with other people. One of the most used social media for socializing is Facebook, with users amounting to over hundred million people around the world. Nowadays, on Facebook, its often found there's hate speech writing being shared at massive pace. Of course an assistance from language expert is a must for identifying hatespeech on Facebook because there's not yet an automatic system that can identify a hatespeech. The system in this research are made using Ensemble Feature and Support Vector Machine. Ensemble Feature is used for combining some of the feature extracted from each writing to ease the process of identifying a hatespeech. Support Vector Machine then used to identifying a hatespeech from a writing based on feature that are combined using ensemble feature. According to the result of testing, we acquired a 70% accuracy for the system so we can conclude that ensemble feature and support vector machine is good to use for identifying hatespeech on social media Facebook.

Keywords: *Hatespeech, Ensemble Feature, Support Vector Machine, Facebook*

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
ABSTRACT	vi
DAFTAR ISI	vii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR.....	xiii
DAFTAR LAMPIRAN	xiv
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	2
1.3 Tujuan	2
1.4 Manfaat.....	2
1.5 Batasan masalah	2
1.6 Sistematika pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Kajian Pustaka	5
2.2 Media Sosial	6
2.2.1 Facebook	7
2.3 <i>Text Mining</i>	7
2.3.1 <i>Case Folding</i>	7
2.3.2 Tokenisasi	8
2.3.3 <i>Filtering</i>	8
2.3.4 <i>Stemming</i>	8
2.4 <i>Feature Extraction</i>	10
2.5 <i>Ensemble Feature</i>	11
2.6 <i>Support Vector Machine</i>	11
2.6.1 <i>Support Vector Machine Linier</i>	13

2.6.2 Sequential Training SVM	15
2.7 Accuracy, Precision, Recall dan F-Measure	17
BAB 3 METODOLOGI	18
3.1 Metodologi Penelitian	18
3.2 Pengumpulan Data	19
3.3 Perancangan	19
3.3.1 Model Perancangan	19
3.4 Implementasi	19
3.5 Pengujian	20
3.6 Analisis	20
3.7 Penarikan Kesimpulan	20
BAB 4 ANALISIS DAN PERANCANGAN SISTEM	21
4.1 Deskripsi Umum Sistem	21
4.2 Pre-processing teks	22
4.2.1 Case Folding	23
4.2.2 Tokenisasi	24
4.2.3 Filtering	25
4.2.4 Stemming	26
4.3 Feature Extraction	27
4.4 Ensemble Feature	31
4.5 Support Vector Machine	32
4.5.1 Perhitungan Kernel	33
4.5.2 Perhitungan Matriks Hessien	34
4.5.3 Perhitungan Sequential Training SVM	35
4.5.4 Perhitungan Bias	38
4.5.5 Perhitungan Identifikasi	40
4.6 Manualisasi Metode Ensemble Feature dan Support Vector Machine	42
4.6.1 Case Folding	43
4.6.2 Tokenisasi	45
4.6.3 Filtering	47
4.6.4 Stemming	50

4.6.5 Manualisasi <i>Feature Extraction</i>	52
4.6.6 Manualisasi <i>Ensemble Feature</i>	54
4.6.7 Manualisasi <i>Support Vector Machine</i>	58
4.7 Perancangan Pengujian	65
4.7.1 Perancangan Pengujian Pengaruh Parameter <i>Support Vector Machine</i>	65
4.7.2 Perancangan Pengujian Pengaruh Jenis <i>Ensemble Feature</i>	66
BAB 5 IMPLEMENTASI	67
5.1 Batasan Implementasi	67
5.2 Implementasi Sistem	67
5.3 <i>Pre-processing</i> teks	69
5.3.1 Implementasi Stopword.....	69
5.3.2 Implementasi Stemming Proses.....	70
5.3.3 Implementasi Tokenisasi.....	71
5.4 Ekstraksi Fitur.....	71
5.4.1 Implementasi Hitung.....	71
5.4.2 Implementasi Import Data.....	74
5.4.3 Implementasi Import Kelas.....	74
5.4.4 Implementasi Hitung Min Max	75
5.5 <i>Ensemble Feature</i>	76
5.5.1 Implementasi Ensemble.....	76
5.6 <i>Support Vector Machine</i>	77
5.6.1 Implementasi Perhitungan Kernel	78
5.6.2 Implementasi Perhitungan Matriks <i>Hessian</i>	78
5.6.3 Implementasi Perhitungan <i>Sequential Training SVM</i> dan Penentuan <i>Support Vector</i>	79
5.6.4 Implementasi Perhitungan <i>bias</i>	81
5.6.5 Implementasi Perhitungan Identifikasi	82
BAB 6 PENGUJIAN	84
6.1 Pengujian Pengaruh Parameter <i>Support Vector Machine</i>	84

6.1.1 Skenario Pengujian Pengaruh Parameter <i>Support Vector Machine</i>	84
6.1.2 Analisis Hasil Pengujian Parameter <i>Support Vector Machine</i>	86
6.2 Pengujian Pengaruh Jenis <i>Ensemble Feature</i>	88
6.2.1 Skenario Pengujian Pengaruh <i>Ensemble Feature</i>	88
6.2.2 Analisis Hasil Pengujian Pengaruh Jenis <i>Ensemble Feature</i>	88
BAB 7 PENUTUP	90
7.1 Kesimpulan.....	90
7.2 Saran	90
DAFTAR PUSTAKA.....	91
LAMPIRAN	93



DAFTAR TABEL

Tabel 2.1 Contoh proses <i>case folding</i>	7
Tabel 2.2 Contoh proses tokenisasi	8
Tabel 2.3 Contoh proses <i>filtering</i>	8
Tabel 2.4 Contoh proses <i>stemming</i>	9
Tabel 2.5 Daftar Fitur	10
Tabel 2.6 Daftar Kernel <i>Support Vector Machine</i>	13
Tabel 4.1 Daftar Fitur	28
Tabel 4.2 Sampel Data Latih.....	42
Tabel 4.3 Sampel Data Uji	43
Tabel 4.4 Data Latih yang mengalami proses <i>case folding</i>	43
Tabel 4.5 Data Uji yang mengalami proses <i>case folding</i>	45
Tabel 4.6 Data Latih yang mengalami proses tokenisasi	45
Tabel 4.7 Data Uji yang mengalami proses tokenisasi.....	47
Tabel 4.8 Data Latih yang mengalami proses <i>filtering</i>	48
Tabel 4.9 Data Uji yang mengalami proses <i>filtering</i>	49
Tabel 4.10 Data Latih yang mengalami proses <i>stemming</i>	50
Tabel 4.11 Data Uji yang mengalami proses <i>stemming</i>	51
Tabel 4.12 Daftar Fitur Khusus Terkait <i>Facebook</i> Hasil <i>Feature Extraction</i>	52
Tabel 4.13 Daftar Fitur Tekstual Hasil <i>Feature Extraction</i>	52
Tabel 4.14 Daftar Fitur Terkait <i>Lexicon</i> Hasil <i>Feature Extraction</i>	53
Tabel 4.15 Hasil <i>Ensemble Feature</i> dari Fitur Terkait <i>Facebook</i> , Tekstual, dan Fitur Terkait <i>Lexicon</i>	54
Tabel 4.16 Hasil <i>Ensemble Feature</i> dari Fitur Tekstual, dan Fitur Terkait <i>Lexicon</i>	55
Tabel 4.17 Hasil <i>Ensemble Feature</i> dari Fitur Terkait <i>Facebook</i> , dan Fitur Terkait <i>Lexicon</i>	55
Tabel 4.18 Hasil <i>Ensemble Feature</i> dari Fitur Terkait <i>Facebook</i> , dan Tekstual	56
Tabel 4.19 Hasil Normalisasi Min-Max Terhadap Fitur 1, 2, 3, 4, dan 5	57
Tabel 4.20 Hasil Normalisasi Min-Max Terhadap Fitur 6, 7, 8, 9, 10. dan 11	57
Tabel 4.21 Fitur dari Data Latih 1, 2, 3, 4, dan 5	58

Tabel 4.22 Fitur dari Data Latih 6, 7, 8, 9, dan 10	58
Tabel 4.23 Fitur dari Data Uji 1, 2, dan 3	59
Tabel 4.24 Hasil Perhitungan Kernel	60
Tabel 4.25 Hasil Perhitungan <i>Matrix Hessian</i>	60
Tabel 4.26 Hasil Perhitungan <i>Error</i>	61
Tabel 4.27 Hasil Perhitungan $\delta\alpha_i$	62
Tabel 4.28 Hasil Perhitungan α Baru	62
Tabel 4.29 Hasil Perhitungan <i>Error</i>	62
Tabel 4.30 Hasil Perhitungan $\delta\alpha_i$	62
Tabel 4.31 Hasil Perhitungan α	63
Tabel 4.32 Hasil Perhitungan <i>Support Vector</i>	63
Tabel 4.33 Proses Perhitungan <i>bias</i>	63
Tabel 4.34 Hasil Perhitungan Identifikasi SVM	64
Tabel 4.35 Perancangan Pengujian Nilai <i>lambda</i> (λ)	65
Tabel 4.36 Perancangan Pengujian Nilai <i>learning rate</i> (γ)	65
Tabel 4.37 Perancangan Pengujian Nilai <i>epsilon</i> (ϵ)	66
Tabel 4.38 Perancangan Pengujian Jenis <i>Ensemble Feature</i>	66
Tabel 5.1 Daftar fungsi sistem identifikasi ujaran kebencian	68
Tabel 6.1 Hasil Pengujian Nilai <i>lambda</i> (λ)	85
Tabel 6.2 Hasil Pengujian Nilai <i>learning rate</i> (γ)	85
Tabel 6.3 Hasil Pengujian Nilai <i>epsilon</i> (ϵ)	85
Tabel 6.4 Hasil Pengujian Pengaruh Jenis <i>Ensemble Feature</i>	88

DAFTAR GAMBAR

Gambar 2.1 Ilustrasi dari <i>Support Vector Machine</i>	12
Gambar 3.1 Tahapan Penelitian.....	18
Gambar 3.2 Model Perancangan Sistem.....	19
Gambar 4.1 Deskripsi Umum Sistem	22
Gambar 4.2 Alur Proses <i>Pre-processing</i> teks	23
Gambar 4.3 Alur Proses <i>Case Folding</i>	24
Gambar 4.4 Alur Proses Tokenisasi.....	25
Gambar 4.5 Alur Proses <i>Filtering</i>	26
Gambar 4.6 Alur Proses <i>Stemming</i>	27
Gambar 4.7 Alur Proses <i>Feature Extraction</i>	30
Gambar 4.8 Alur Proses <i>Ensemble Feature</i>	31
Gambar 4.9 Alur Proses <i>Support Vector Machine</i>	32
Gambar 4.10 Alur Proses Perhitungan Kernel	33
Gambar 4.11 Alur Proses Perhitungan Matrix Hessian.....	34
Gambar 4.12 Alur Proses Perhitungan <i>Sequential Training SVM</i>	37
Gambar 4.13 Alur Proses Perhitungan <i>Bias</i>	40
Gambar 4.14 Alur Proses Perhitungan Identifikasi.....	41
Gambar 5.1 Cuplikan isi <i>file</i> negatif.txt.....	73
Gambar 5.2 Cuplikan isi <i>file</i> positif.txt	73
Gambar 6.1 Contoh Hasil Pengujian	84
Gambar 6.2 Hasil pengujian pengaruh <i>lambda</i>	86
Gambar 6.3 Hasil pengujian pengaruh <i>learning rate</i>	87
Gambar 6.4 Hasil pengujian pengaruh <i>epsilon</i>	87
Gambar 6.5 Hasil pengujian pengaruh jenis <i>Ensemble Feature</i>	89

DAFTAR LAMPIRAN

LAMPIRAN 1 Tabel Kalimat Data Latih 93

LAMPIRAN 2 Tabel Kalimat Data Uji 100

LAMPIRAN 3 Tabel Fitur Facebook dan Kelas Data Latih..... 102

LAMPIRAN 4 Tabel Fitur Facebook dan Kelas Data Uji 106

LAMPIRAN 5 Surat Kesediaan Pakar 108



BAB 1

PENDAHULUAN

1.1 Latar belakang

Ujaran kebencian atau *hatespeech* merupakan sebuah bentuk ujaran yang menargetkan karakteristik grup tertentu, seperti asal etnis, agama atau kepercayaan, jenis kelamin, atau orientasi seksual (Warner & Hirschberg, 2012). Media sosial sendiri merupakan sebuah media *online* yang membuat penggunanya dapat melakukan interaksi sosial antar manusia. Media sosial tidak jarang digunakan untuk menyebarkan berita-berita kepada publik untuk mendapatkan cakupan yang lebih luas. Akan tetapi tidak jarang juga disalahgunakan untuk menyebarkan berita bohong, fitnah, atau ujaran kebencian.

Pada umumnya, ujaran kebencian sendiri sering muncul di linimasa media sosial kita tanpa dikehendaki, selain itu para pelaku ujaran kebencian biasanya dilakukan tindakan setelah ada laporan dari pihak lain. Hal ini tentu sangat merugikan mengingat cepatnya konten di media sosial dapat menyebar dalam hitungan waktu. Diperlukan tindakan preventif berupa identifikasi ujaran kebencian untuk mencegah penyebaran ujaran kebencian, serta untuk membantu pengguna melakukan *filtering* terhadap linimasa media sosialnya.

Facebook adalah sebuah layanan media sosial yang dikembangkan pada tahun 2004 oleh mahasiswa Harvard Mark Zuckerberg, yang memungkinkan pengguna untuk menambahkan teman, berkirim pesan, dan memperbaharui profil personal untuk memberitahu teman dan kolega tentang mereka. Pengguna *Facebook* sendiri dapat membuat dan mengikuti grup virtual, mengembangkan aplikasi, *host* konten, dan mempelajari tentang minat, hobi, dan status relasi satu sama lain melalui profil *online* pengguna (Quan-Haase & Young, 2011).

Pada tahun 2015, penelitian tentang pendeteksian hate speech (ujaran kebencian) dilakukan oleh dua mahasiswa Kolombia dari *Columbia University* yakni Warner & Hirschberg. Penelitian ini menggunakan metode *feature template* untuk melakukan *pre-processing* data dari korpus yang kemudian digabungkan dengan penggunaan *Support Vector Machine* (SVM) sebagai *classifier* dengan hasil akurasi 94% untuk *anti-semitic speech*. Peneliti menggunakan fitur berupa tulisan pada postingan *news group posts* di *Yahoo!*. Peneliti juga melakukan penggabungan antara classifier SVM dengan *brown clusters*. Akan tetapi penggabungan antara SVM dengan *brown cluster* menghasilkan akurasi yang kurang dibandingkan dengan *feature template*.

Pada penelitian lain yang dilakukan oleh Zia et. al. pada tahun 2017 mengenai identifikasi ujaran kebencian pada twitter. Peneliti melakukan

komparasi antara metode SVM, kNN, dan Naïve-Bayes dengan menggunakan fitur *Bag-of-Words*(BoW). Dari hasil penelitian ini didapatkan hasil metode SVM menghasilkan nilai dari akurasi, *Recall*, dan *F-Measure* lebih baik daripada kedua metode lainnya.

Penelitian yang dilakukan oleh Siddiqua et. al. di tahun 2016 mengenai penggunaan ensemble feature untuk melakukan *Sentiment Analysis* pada *microblog*. Peneliti menggunakan metode ekstraksi fitur berupa sekumpulan fitur yang dikelompokkan menjadi beberapa kategori berbeda, termasuk fitur spesifik *twitter*, fitur tekstual, fitur *parts-of-speech* (POS), dan fitur berdasar *lexicon*(Siddiqua et. al., 2016).

Sistem diharapkan mampu untuk mengidentifikasi ujaran kebencian atau bukan ujaran kebencian dalam sebuah post di media sosial *Facebook*. Berdasarkan latar belakang tersebut, akan diimplementasikan metode *Ensemble Feature* dan *Support Vector Machine* untuk mengidentifikasi ujaran kebencian pada media sosial *Facebook*.

1.2 Rumusan masalah

Berdasarkan masalah yang telah dijelaskan, rumusan masalah dalam skripsi ini adalah:

1. Bagaimana pengaruh parameter *Support Vector Machine* terhadap identifikasi ujaran kebencian pada Facebook?
2. Bagaimana pengaruh metode *Ensemble Feature* pada identifikasi ujaran kebencian pada Facebook menggunakan *Support Vector Machine*?

1.3 Tujuan

Tujuan dari penulisan skripsi ini adalah untuk melakukan identifikasi ujaran kebencian pada *Facebook* dengan metode *Ensemble Feature* dan *Support Vector Machine*.

1.4 Manfaat

Manfaat penulisan skripsi ini bagi penulis sendiri adalah sebagai sebuah sarana untuk menerapkan materi yang diperoleh selama proses perkuliahan. Selain itu penulis juga dapat belajar mengenai pengembangan pemrosesan bahasa alami. Manfaat bagi pengguna aplikasi ini adalah mendeteksi ujaran kebencian pada *Facebook*.

1.5 Batasan masalah

Agar pembahasan masalah menjadi lebih fokus, berdasarkan rumusan masalah, batasan-batasan yang perlu diterapkan pada skripsi ini adalah:

1. Data latih berasal dari 75 *post* berisi ujaran kebencian dan 75 *post* berisi ujaran bukan kebencian.
2. Data merupakan data berbasis teks yang diambil dari berbagai laman di *Facebook*.
3. Algoritme yang digunakan adalah *Ensemble Feature* dan *Support Vector Machine*.
4. Satu ujaran kebencian adalah satu kalimat berisikan kriteria ujaran kebencian

1.6 Sistematika pembahasan

Dalam proses menyusun skripsi ini terdapat format penulisan dengan struktur sebagai berikut:

BAB I PENDAHULUAN

Bab Pendahuluan berisi latar belakang mengenai ujaran kebencian (*hatespeech*) dan media sosial *Facebook*. Dalam bab ini terdapat juga rumusan masalah yang menjelaskan apa saja yang dapat diuraikan mengenai penerapan algoritme *Ensemble Feature* dan *Support Vector Machine*, serta tujuan penulisan yang akan dicapai penulis dengan dikembangkannya aplikasi ini. Bab ini juga berisi batasan masalah dari masalah yang dibahas dalam penerapan algoritme *Ensemble Feature* dan *Support Vector Machine* pada identifikasi ujaran kebencian. Selain itu, bab ini juga berisi manfaat dari penulisan skripsi ini.

BAB II LANDASAN KEPUSTAKAAN

Bab Landasan Kepustakaan berisikan penjelasan dasar dari beberapa teori yang digunakan dan beberapa penelitian sebelumnya yang memiliki hubungan dengan topik-topik pada skripsi Identifikasi ujaran kebencian pada *Facebook* dengan metode *Ensemble Feature* dan *Support Vector Machine*.

BAB III METODOLOGI

Bab Metodologi berisikan tahapan yang harus dilakukan dalam proses implementasi identifikasi ujaran kebencian pada *Facebook* dengan metode *Ensemble Feature* dan *Support Vector Machine*. Pada bab ini juga diuraikan mengenai metode yang diimplementasikan pada Identifikasi Ujaran Kebencian pada *Facebook* dengan metode *Ensemble Feature* dan *Support Vector Machine*.

BAB IV ANALISIS DAN PERANCANGAN

Bab Perancangan berisi analisis mengenai hal-hal yang diperlukan untuk melanjutkan ke perancangan sistem, selain analisis terdapat juga perancangan dari sistem yang akan dibuat.

BAB V IMPLEMENTASI

Bab Implementasi berisi tahapan dan strategi untuk mengimplementasikan rancangan yang telah dibuat sebelumnya serta berisi pembahasan mengenai fungsi dari tiap bagian sistem yang dibuat.

BAB VI PENGUJIAN

Bab Pengujian berisikan proses dan hasil dari pengujian yang dilaksanakan terhadap data dengan menggunakan algoritme *Ensemble Feature* dan *Support Vector Machine* untuk mengidentifikasi ujaran kebencian.

BAB VII KESIMPULAN DAN PENUTUP

Bab ini membahas mengenai kesimpulan yang didapat dari hasil pembuatan sistem identifikasi ujaran kebencian pada Facebook dengan metode *Ensemble Feature* dan *Support Vector Machine* dan saran-saran untuk mengembangkan sistem

BAB 2

LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Pada tahun 2003, Tsymbal et. al., melakukan sebuah penelitian mengenai pemilihan fitur secara *ensemble* dengan *classifier* Bayes sederhana. Untuk tiap *dataset* dibuat tiga puluh *test run* dari EFS_SBC. Dalam tiap *test run* dataset dibagi menjadi set data latih TrS, set validasi data VS, dan set data uji TS oleh sampling acak bertingkat. Algoritme yang digunakan oleh penulis adalah algoritme *Ensemble Feature Selection with the Simple Bayesian Classification*(EFS_SBC), EFS_SBC membangun *ensemble* dari *classifier* Bayes sederhana dalam *sub-space* acak dan menggunakan *hill-climbing* dalam sebuah siklus yang telah diperbaiki untuk meningkatkan akurasi dan keragaman dari *classifier* dasar dalam *subspace* acak. Hasil dari penelitian ini ada dua poin utama yaitu: (1) Derajat kepentingan dari akurasi dan keberagaman ketika membangun *ensemble* berbeda untuk tiap *dataset*, dan (2) Integrasi dinamis memiliki akurasi yang lebih baik daripada statis,

Pada tahun 2007, Warner & Hirschberg melakukan penelitian mengenai pendeteksian *hate speech* pada *World Wide Web*. Data yang mereka gunakan berasal dari *Yahoo! & American Jewish Congress*. Data yang diberikan berupa *posting* pada *news group* dan *pointer* yang menunjuk ke *website* yang teridentifikasi sebagai ofensif. Untuk proses pendeteksian *hate speech* ini mereka menggunakan strategi *template-based* dengan menggunakan SVM. Hasilnya, penggunaan *template bigram* dan *trigram* mengurangi kinerja dari *classifier*.

Kemudian, pada tahun 2008, Saeys et al., melakukan penelitian untuk teknik pemilihan fitur dengan menggunakan teknik pemilihan fitur secara *Ensemble*. Data yang digunakan berasal dari domain bioinformatika dan biomedika dan dibagi menjadi dua bagian yaitu: *dataset microarray* (MA) dan *dataset spektrometri massa* (MS). Untuk tiap domain yang disertakan, biasanya berisikan ribuan fitur dan puluhan *instance* pada *dataset microarray*, dan sampai lima belas ribu fitur dan ratusan *instance* pada *dataset spektrometri massa*.

Pada tahun 2013, Huang dan Zhang melakukan penelitian pada *SVM Ensemble* yang digunakan untuk klasifikasi citra penginderaan jauh. Penulis penggabungan terhadap beberapa fitur dari *SVM* berupa fitur spektral dan fitur spasial pada tingkat objek dan piksel. Fitur yang digunakan adalah fitur matriks kemunculan tingkat abu-abu, profil morfologi diferensial, dan indeks kompleksitas urban. Selain itu digunakan tiga algoritme untuk mengintegrasikan *SVM* multi fitur yaitu menggunakan *certainty voting*, *probabilistic fusion*, dan *object-based semantic approach*. Hasil dari penelitian mereka adalah dengan

penggunaan algoritme *C-Voting* dan *P-Fusion* terbukti efektif untuk penggabungan multi fitur, penelitian yang dilakukan oleh penulis menunjukkan bahwa *knowledge-based rules* lebih krusial untuk interpretasi akurat dari gambar beresolusi tinggi.

Lalu, pada penelitian yang dilakukan oleh Siddiqua et. al. di tahun 2016 mengenai penggunaan *ensemble feature* untuk melakukan *Sentiment Analysis* pada microblog. Peneliti menggunakan metode untuk *Supervised Feature Selection* dengan cara mengelompokkan beberapa fitur menjadi beberapa kategori berbeda, termasuk fitur spesifik *social media*, fitur tekstual, fitur *parts-of-speech* (POS), fitur berdasar lexicon, dan *bag of words* (BoW). Pada awalnya dilakukan seleksi fitur dan melakukan perankingan fitur berdasar kepentingan fitur setelah itu dilakukan analisa sentiment dengan menggunakan *lexicon*. Dengan menggunakan beberapa *lexicon sentiment* untuk data latih lebih efektif daripada menggunakan *dataset* yang besar.

Pada penelitian yang dilakukan oleh Zia et. al. pada tahun 2017 tentang identifikasi ujaran kebencian pada twitter, dilakukan perbandingan antara tiga metode identifikasi yaitu SVM, Naïve-Bayes, dan kNN. Penelitian ini menggunakan fitur *Bag-of-Words*(BoW) dalam proses identifikasinya. Hasil dari penelitian ini menunjukkan bahwa SVM memiliki akurasi yang jauh lebih baik dari Naïve-Bayes dan kNN.

Pada penelitian ini, saya melakukan penerapan *Ensemble Feature* dan SVM untuk mengidentifikasi ujaran kebencian pada Facebook. Setelah itu akan dilakukan pengujian untuk menghitung tingkat akurasi dari penggunaan *Ensemble Feature* dan SVM . Diharapkan penggunaan *Ensemble Feature* dan SVM menghasilkan akurasi yang lebih baik dalam pengidentifikasian ujaran kebencian dibanding penggunaan SVM saja.

2.2 Media Sosial

Media sosial merupakan sebuah media yang membantu mendukung terjadinya proses interaksi sosial antar manusia secara *online* melalui internet. Media sosial menggunakan teknologi berbasis web yang mampu untuk mengubah data komunikasi menjadi percakapan antar masing-masing pengguna media sosial. Pengguna media sosial yang semakin bertambah dikarenakan kegiatan interaksi yang lebih cepat melalui media sosial dan tidak terikat batasan wilayah.

Media sosial dikarakterisikan oleh interaktivitasnya – partisipan dengan bebas mengirim, menerima, dan memproses konten untuk digunakan oleh partisipan lainnya. Layanan media sosial termasuk jaringan sosial, pembuatan konten, distribusi layanan dan *website* yang dibangun secara kolektif oleh pengguna(contohnya *Wikipedia*), layanan berbagi video dan foto(contohnya *Youtube* dan *Flickr*), dunia virtual (*Second Life*), dan *website* bertipe diari (blog).

Dari perspektif korporasi, layanan media sosial yang paling menarik dan paling populer termasuk layanan jaringan sosial terbesar didunia, Facebook, MySpace yang terfokus pada musik dan hiburan, LinkedIn yang berorientasi karir, dan layanan jaringan Twitter, yang memungkinkan anggota mengirim pesan singkat melalui komputer dan perangkat bergerak. (Aula, Pekka,2010).

2.2.1 Facebook

Facebook adalah salah satu media sosial yang paling populer untuk digunakan, hal ini dikarenakan fitur-fitur dari Facebook yang mendukung untuk terjadinya interaksi antar orang secara *online* melalui fitur *post*, *comment* dan *share*. Untuk fitur *post* sendiri pengguna dapat melakukan *post* menggunakan teks atau dapat menggunakan gambar, untuk fitur *share* pengguna dapat membagikan *post* dari orang lain ke dalam lingkaran teman mereka sendiri.

Facebook sendiri awalnya dimaksudkan sebagai sarana berbagi informasi, bersosialisasi, dan menambahkan teman atau kolega. Tetapi akhir-akhir ini *Facebook* seringkali di salah gunakan oleh sebagian orang untuk menyebarkan ujaran kebencian dan atau hoax. *Facebook* juga sudah melakukan beberapa tindakan untuk akun penyebar hoax tetapi untuk penindakan biasanya hanya berdasarkan pada laporan beberapa pengguna saja, yang mana dampak hoax itu sudah menyebar luas.

2.3 Text Mining

Text mining adalah metode untuk mengolah sebuah teks atau dokumen yang tidak diurutkan sebelumnya. *Text mining* atau pemrosesan teks muncul pertama kalinya pada pertengahan tahun 1980 dan pemrosesan teks sendiri sering dipergunakan dalam beberapa disiplin ilmu lain antara lain: *Natural Language Processing*, *Information Retrieval*, dan *Data Mining*. Dalam melakukan pemrosesan teks terdapat beberapa proses atau tahapan yaitu: *case folding*, *tokenisasi*, *filtering*, dan *stemming*.

2.3.1 Case Folding

Case Folding merupakan proses atau tahapan untuk melakukan perubahan terhadap huruf dalam dokumen menjadi huruf kecil, dan karakter lain di luar huruf seperti angka dan tanda baca dihilangkan (Arifin & Purnama, 2012). Untuk contoh dari sebuah proses *Case Folding* dapat dilihat pada Tabel 2.1.

Tabel 2.1 Contoh proses *case folding*

Data Input	Proses <i>Case Folding</i>
"Jika ada orang yang demikian	"jika ada orang yang demikian

gampangnya menuduh orang lain sebagai pecandu PCC dan FLAKKA, maka mungkin dia... Ah.... Tak tega aku meneruskannya bang!"	gampangnya menuduh orang lain sebagai pecandu pcc dan flakka, maka mungkin dia... ah.... tak tega aku meneruskannya bang!"
--	--

2.3.2 Tokenisasi

Dalam tahap tokenisasi sebuah kalimat di pilah berdasarkan pada kata-kata yang menyusunnya (Arifin & Purnama, 2012). Untuk contoh dari sebuah proses tokenisasi ditunjukkan pada Tabel 2.2.

Tabel 2.2 Contoh proses tokenisasi

Data Input	Proses tokenisasi
"jika ada orang yang demikian gampangnya menuduh orang lain sebagai pecandu pcc dan flakka, maka mungkin dia... ah.... tak tega aku meneruskannya bang!"	['jika', 'ada', 'orang', 'yang', 'demikian', 'gampangnya', 'menuduh', 'orang', 'lain', 'sebagai', 'pecandu', 'pcc', 'dan', 'flakka,', 'maka', 'mungkin', 'dia...', 'ah....', 'tak', 'tega', 'aku', 'meneruskannya', 'bang!']

2.3.3 Filtering

Filtering merupakan tahap membuang kata-kata yang tidak deskriptif dengan menggunakan *stoplist*, seperti kata 'ke', 'dari', 'yang', 'sedang', 'di', 'ini', dst. (Arifin & Purnama, 2012). Untuk contoh dari sebuah tahap *filtering* ditunjukkan pada Tabel 2.3.

Tabel 2.3 Contoh proses filtering

Data Input	Proses filtering
['jika', 'ada', 'orang', 'yang', 'demikian', 'gampangnya', 'menuduh', 'orang', 'lain', 'sebagai', 'pecandu', 'pcc', 'dan', 'flakka,', 'maka', 'mungkin', 'dia...', 'ah....', 'tak', 'tega', 'aku', 'meneruskannya', 'bang!']	['jika', 'orang', 'demikian', 'gampangnya', 'menuduh', 'orang', 'sebagai', 'pecandu', 'pcc', 'flakka', 'mungkin', 'ah', 'tak', 'tega', 'aku', 'meneruskannya', 'bang']

2.3.4 Stemming

Stemming adalah sebuah metode untuk mencari kata dasar dari kata berimbuhan dengan menggunakan aturan morfologi (Asian, 2007). Tahapan proses ini akan mengubah kata berimbuhan menjadi kata dasar untuk

mengoptimalkan proses *Feature Extraction*. Untuk contoh dari sebuah proses *stemming* dapat dilihat pada Tabel 2.4.

Tabel 2.4 Contoh proses *stemming*

Data Input	Proses <i>stemming</i>
['jika', 'orang', 'demikian', 'gampang', 'menuduh', 'orang', 'sebagai', 'pecandu', 'pcc', 'flakka', 'mungkin', 'ah', 'tak', 'tega', 'aku', 'meneruskannya', 'bang']	['jika', 'orang', 'demikian', 'gampang', 'tuduh', 'orang', 'bagai', 'candu', 'pcc', 'flakka', 'mungkin', 'ah', 'tak', 'tega', 'aku', 'terus', 'bang']

Beberapa aturan Bahasa diterapkan untuk menghapus imbuhan-imbuhan tersebut. Aturan ini dibuat karena banyaknya permasalahan dalam proses *stemming* kata bahasa Indonesia (Asian, 2007), antara lain:

1. Imbuhan yang kompleks pada Bahasa Indonesia
2. *Word-Sense Ambiguity* (Ambiguitas Kata), yakni satu kata yang memiliki dua arti / makna yang berbeda / bermakna ganda dan berasal dari dua kata berbeda
3. *Overstemming*, atau proses *stemming* berlebih pada suatu kata.
4. *Understemming*, atau proses *stemming* yang tidak sempurna sehingga menghasilkan kata dasar yang tidak sesuai dengan kata dasar yang dimaksud.
5. Berbeda-bedanya opini tiap orang dalam melakukan stem manual terhadap Bahasa Indonesia
6. Kata dengan bentuk jamak
7. Kata hasil serapan bahasa asing
8. Kesalahan dalam penulisan kata
9. Akronim, yaitu kependekan yang biasanya berupa gabungan huruf atau suku kata atau bagian lain yang ditulis dan dilafalkan sebagai kata yang wajar.
10. *Proper Noun* atau nama benda,

Terdapat banyak permasalahan dari proses *stemming* tersebut, diperlukannya sebuah algoritme *stemmer* yang tepat untuk mengatasi permasalahan tersebut. Terdapat beberapa algoritme *stemmer* yang ditemukan pada penelitian *text mining*. Salah satu di antaranya adalah *stemmer* Sastrawi. *Stemmer* Sastrawi adalah sebuah library *stemmer* yang didesain untuk dapat digunakan secara mudah dan mampu untuk menyelesaikan masalah-masalah

dias. *Stemmer* Sastrawi menggunakan beberapa algoritme antara lain algoritme Nazief Adriani, yang kemudian ditingkatkan dengan menggunakan algoritme *Confix Stripping*, setelah itu ditingkatkan lagi menggunakan algoritme *Enhance Confix Stripping*, dan ditingkatkan dengan algoritme *Modified Enhance Confix Stripping*. Dengan penggunaan algoritme tersebut, Sastrawi *stemmer* mampu untuk menyelesaikan sebagian besar permasalahan diatas.

Sastrawi *stemmer* memiliki sekitar 40 aturan untuk pemenggalan kata. Namun, seluruh permasalahan diatas tidak dapat terpecahkan seluruhnya dengan *stemmer* ini. Permasalahan yang belum terselesaikan adalah sebagai berikut:

1. *Word-Sense Ambiguity*
2. Berbedanya opini tiap orang dalam melakukan *stemming* manual
3. Akronim dan *Proper Noun*

2.4 Feature Extraction

Untuk *Feature Extraction*, dalam skripsi ini digunakan *Feature Extraction* seperti yang digunakan oleh Siddiqua et. al., yaitu berupa sekumpulan fitur yang dikelompokkan menjadi beberapa kategori berbeda, termasuk fitur spesifik, fitur tekstual, dan fitur berdasar lexicon (Siddiqua et. al., 2016).

Untuk penggunaan *Feature Extraction* dalam skripsi ini telah dibuatkan Tabel 2.5 untuk menspesifikasikan fitur mana saja yang akan digunakan untuk proses identifikasi

Tabel 2.5 Daftar Fitur

Kategori Fitur	Fitur	Keterangan Fitur
Fitur khusus terkait <i>Facebook</i>	F1	Jumlah <i>like react</i> pada <i>post</i>
	F2	Jumlah <i>angry react</i> pada <i>post</i>
	F3	Jumlah komentar pada <i>post</i>
	F4	Jumlah <i>share post</i>
Fitur Tekstual	F5	Jumlah kata dalam <i>post</i>
	F6	Jumlah tanda seru dalam <i>post</i>
	F7	Jumlah tanda tanya dalam <i>post</i>

Fitur Terkait Lexicon	F8	Jumlah kata dengan huruf kapital
	F9	Jumlah kata dengan huruf kecil
	F10	Kata berkonotasi Negatif
	F11	Kata berkonotasi Positif

2.5 Ensemble Feature

Ensemble Feature merupakan sebuah metode *Ensemble* yang menggabungkan beberapa fitur sekaligus yang kemudian akan dimasukkan ke dalam *classifier* untuk melakukan proses identifikasi. *Ensemble Feature* ini sendiri memiliki beberapa varian lain seperti model yang digunakan oleh Serbes et. al., untuk meningkatkan akurasi rata-rata dan kemampuan generalisasi dari model yang mereka bangun berdasarkan bukti dari Hansen & Salamon: jika tiap anggota dari *ensemble*, contohnya sub bagian fitur, bisa mendapatkan jawaban yang benar lebih dari setengah waktu yang diperlukan, dan jika respon dari tiap anggota bersifat independen, kemungkinan kesalahan dengan strategi *majority voting* akan secara monoton berkurang dengan jumlah anggota yang lebih banyak (Serbes et. al., 2011).

Dalam penelitian ini akan digunakan beberapa kategori fitur dalam pembuatan *ensemble feature* antara lain fitur khusus terkait *Facebook*, tekstual, dan *lexicon*. Dari ketiga fitur ini kemudian akan diuraikan lagi menjadi beberapa fitur-fitur individual yaitu: fitur jumlah *like* dan *angry react* pada *post* serta jumlah komentar dan *share* pada postingan untuk kategori fitur khusus terkait *Facebook*, fitur jumlah kata, tanda seru, dan tanda tanya dalam *post* beserta jumlah kata dengan huruf capital dan huruf kecil dalam *post* untuk kategori fitur tekstual, dan fitur kata berkonotasi negatif dan positif untuk kategori fitur *lexicon*.

2.6 Support Vector Machine

Support Vector Machine adalah sebuah algoritme pengklasifikasian yang menggabungkan tiga ide, yaitu: teknik penyelesaian dari *hyperplane* optimal(yang memungkinkan untuk ekspansi dari vektor solusi pada *support vector*), ide mengenai konvolusi *dot-product*(yang menambah bidang solusi dari

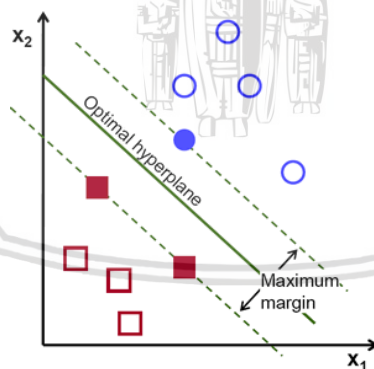
linier ke non-linier) dan notasi dari *soft margin* (yang memungkinkan kesalahan pada set data latih) (Cortes & Vapnik, 1995).

Berdasarkan hasil penelitian yang dilakukan oleh (Zia et. al.,2017). SVM sendiri memiliki beberapa keunggulan di bandingkan dengan Naïve-Bayes dan kNN antara lain:

- SVM memiliki mekanisme *over-fitting* internal yang baik sehingga dapat memproses data dengan dimensionalitas yang tinggi
- SVM bergantung pada jumlah contoh yang relatif kecil
- SVM dapat menghasilkan akurasi yang berdasarkan pada basis teoritis, bahkan ketika datanya tidak monoton dan tidak terpisah secara linier
- SVM efektif dalam ruang dimensi tinggi, bahkan efektif pada kasus ketika jumlah dimensi lebih banyak dari jumlah sampel ‘

Pada *support vector machine*, klasifikasi dilakukan dengan mencari *hyperplane* yang memiliki fungsi untuk memisahkan dua kelas data yang berbeda. SVM dapat bekerja secara akurat pada data dengan dimensi tinggi dengan cara mempergunakan *kernel trick*. SVM menggunakan beberapa titik data yang dipilih sebagai *support vector* untuk membentuk model yang digunakan dalam klasifikasi.

Berikut merupakan ilustrasi atau gambaran dari Support Vector Machine yang berupa titik data (*vector*), *support vector*, kelas data, dan *hyperplane* pada Gambar 2.1



Gambar 2.1 Ilustrasi dari Support Vector Machine

Pemodelan dari ilustrasi *Support Vector Machine* diatas adalah sebagai berikut:

1. Titik data: $x_i = \{x_1, x_2, \dots, x_n\} \in R^n$
2. Kelas data: $y_i \in \{-1, +1\}$
3. Pasangan kelas data dan data: $\{(x_i, y_i)\}_{i=1}^N$

4. Fungsi keputusan klasifikasi $\text{sign}(h(x))$: $f(x) = w \cdot x + b$ atau $f(x) = \sum_{i=1}^m a_i y_i K(x, x_i) + b$

Keterangan:

- N = jumlah data
- n = jumlah fitur / dimensi dari data
- w = parameter *hyperplane* (garis tegak lurus antara titik-titik support vector)
- x = titik data input *Support Vector Machine*
- a_i = bobot tiap titik data
- $K(x, x_i)$ = fungsi kernel
- b = nilai *bias* dari parameter *Support Vector Machine*

Beberapa fungsi kernel pada *Support Vector Machine* yang kemudian akan dipergunakan untuk *kernel trick* pada Tabel 2.6

Tabel 2.6 Daftar Kernel *Support Vector Machine*

No	Nama Kernel	Definisi Fungsi
1	Linier	$K(x, y) = x \cdot y$
2	Polinomial dengan pangkat d	$K(x, y) = (x \cdot y)^d$
3	Polinomial dengan pangkat sampai d	$K(x, y) = (x \cdot y + c)^d$
4	<i>Gaussian Radial Basis Function</i>	$K(x, y) = \exp\left(\frac{- x - y ^2}{2a^2}\right)$
5	<i>Sigmoid (Hyperbolic Tangent)</i>	$K(x, y) = \tanh(\sigma(x \cdot y) + c)$
6	<i>Additive</i>	$K(x, y) = \sum_{i=1}^n K(x_i, y_i)$
7	Invers Multi Kuadratik	$K(x, y) = \frac{1}{\sqrt{ x - y ^2 + c^2}}$

2.6.1 *Support Vector Machine Linier*

Algoritme asli dari *maximum-margin hyperplane* yang diajukan oleh Vapnik pada tahun 1963 menghasilkan sebuah *classifier* linier yang kemudian disebut sebagai *SVM linier*. *SVM Linier* terbagi menjadi dua jenis, yaitu untuk masalah yang *separable* dan *non-separable*. *SVM Linier* melakukan klasifikasi terhadap himpunan data latih dengan cara berpasangan dari tiap kelas.

Klasifikasi *linier* merupakan cara memisahkan data secara *linier*, yaitu data mampu dipisahkan dengan menggunakan bidang pemisah sesuai dengan kelas dan dengan margin yang paling besar. Bidang pemisah pertama membatasi kelas positif dan bidang pemisah kedua membatasi kelas negatif dari himpunan data, sehingga diperoleh fungsi pembatas bidang untuk tiap kelas dalam Persamaan 2.1.

$$(w \cdot x_i + b) \geq +1 \text{ untuk } y_i = +1 \quad (2.1)$$

$$(w \cdot x_i + b) \leq -1 \text{ untuk } y_i = -1$$

Dengan variabel w sebagai bidang antara bidang pemisah terhadap pusat koordinat dan b merupakan posisi relatif bidang terhadap pusat koordinat serta x_i sebagai titik dalam dataset dan y_i sebagai kelas data. Setelah itu dilakukan pencarian margin terbesar dengan cara memaksimalkan fungsi $\frac{1}{w}$ atau dengan meminimalkan fungsi $\|w\|^2$. Untuk mendapatkan *hyperplane* atau pemisah untuk kedua kelas digunakan Persamaan 2.2.

$$y_i(w \cdot x_i + b) \geq 1, i = 1, 2, 3, \dots, n \quad (2.2)$$

Hal ini kemudian dapat dijadikan *Quadratic Programming* dengan mencari titik minimal dari kedua fungsi tersebut. Untuk mendapatkan *hyperplane* dengan margin terbesar dapat diubah menjadi Persamaan 2.3.

$$\text{Min} = \frac{1}{2} \|w\|^2 \quad (2.3)$$

Dengan syarat $y_i(w \cdot x_i + b) \geq 1$ Persamaan kemudian dipergunakan untuk mengatasi masalah optimasi yang dapat diatasi dengan fungsi *Lagrange*. Fungsi *Lagrange* yang digunakan yaitu fungsi *Lagrange multiplier* dengan Persamaan 2.3 sehingga menghasilkan Persamaan 2.4.

$$\text{Min } L(w, b, a) = \frac{1}{2} \|w\|^2 \sum_{i=1}^n a_i [y_i(w \cdot x_i + b) - 1] \quad (2.4)$$

Dengan variabel a_i sebagai nilai bobot untuk tiap data. Variabel ini memiliki nilai nol atau positif ($a_i \geq 0$), dengan meminimalkan nilai L terhadap w dan b , maka untuk melakukan perhitungan terhadap variabel w dan b digunakanlah Persamaan 2.5.

$$w = \sum_{i=1}^n a_i y_i x_i \text{ dan } b = -\frac{1}{2} (w \cdot x^+ + w \cdot x^-) \quad (2.5)$$

x^+ dan x^- masing-masing merupakan nilai dari *support vector* dari kelas positif dan kelas negatif. Persamaan 2.5 kemudian dapat dimodifikasi sehingga menjadi sebuah fungsi untuk mencari *hyperplane* atau bidang pemisah terbaik yang dapat dilihat pada Persamaan 2.6.

$$\max \sum_{i=1}^n a_i - \frac{1}{2} a_i a_j y_i y_j x_i x_j \quad (2.6)$$

dengan $a_i \geq 0 (i = 1, 2, \dots, n)$ dan $\sum_{i,j=1}^n a_i y_i = 0$

Dari hasil perhitungan Persamaan 2.6 diperoleh nilai α bernilai positif, yang akan digunakan untuk menemukan nilai W . Dari setiap data latih, diperoleh nilai α_i . Dari nilai α_i kemudian di cari *Support vector* yang merupakan data latih yang memiliki nilai $\alpha_i > 0$, sedangkan yang lainnya memiliki nilai $\alpha_i = 0$. Sehingga, fungsi keputusan dari *Support vector machine* hanya dipengaruhi oleh nilai dari *support vector*. Untuk memperoleh hasil klasifikasi yang optimal digunakan Persamaan 2.7 . Hasil klasifikasi kemudian diperoleh menggunakan Persamaan 2.8.

$$f(x) = \sum_{i=1}^n a_i y_i x_i \cdot x + b \quad (2.7)$$

$$f(x) = \sum_{i=1}^n (w \cdot x + b)$$

$$\text{Fungsi Klasifikasi} = \text{sign}(f(x)) \quad (2.8)$$

2.6.2 Sequential Training SVM

Sequential training SVM adalah sebuah metode hasil pengembangan dari *Support Vector Machine* yang digunakan untuk memberikan solusi yang lebih optimal, meningkatkan kecepatan dari proses iterasi, dan memberikan urutan dengan lebih cepat jika dibandingkan dengan pemberian solusi dari metode *Support Vector Machine* pada umumnya. Metode ini pertama kali diperkenalkan oleh Vijayakumar dan Wu(1999).

Sedangkan langkah-langkah atau tahapan dari proses *sequential training SVM* adalah:

1. Inialisasi beberapa variabel yang diperlukan dalam proses perhitungan, yaitu λ , γ (*learning rate*), C (*slack variable*), dan ϵ (*epsilon*).
2. Menginisialisasi $\alpha_1 = 0$ dan melakukan perhitungan matriks Hessian D_{ij} dengan menggunakan Persamaan 2.9.

$$D_{ij} = y_i y_j (K(x_i, x_j) + \lambda^2) \quad (2.9)$$

Keterangan:

- D = Matriks Hessian
 - y = kelas data
 - x = data
 - K = fungsi kernel
 - λ = variabel skalar
3. Lakukan langkah berikut ini dari $i=1$ sampai N (banyak data).

$$a. E_i = \sum_{i=0}^N a_i D_{ij} \quad (2.10)$$

$$b. \delta\alpha_i = \min\{\max[\gamma(1 - E_i), -a_i], C - a_i\} \quad (2.11)$$

$$c. \alpha_i = a_i + \delta\alpha_i \quad (2.12)$$

Keterangan:

- E = Error
- $\delta\alpha$ = perubahan α
- γ = *learning rate*
- C = *slack variable*
- α = *Lagrange Multiplier*

4. Lakukan langkah ketiga hingga mencapai $\text{Max}(|\delta\alpha|) < \epsilon$
5. Dari hasil proses diatas kemudian diperoleh nilai α dan *support vector*, yang merupakan titik data yang memiliki nilai $\alpha > 0$. Setelahnya, dihitung nilai *bias* b yang diperoleh dengan Persamaan:

$$b = -\frac{1}{2}(\sum_{i=0}^N \alpha_i \gamma_i K(x_i, x^+) + \sum_{i=0}^N \alpha_i \gamma_i K(x_i, x^-)) \quad (2.13)$$

Keterangan:

- b = bias
- α = *Lagrange Multiplier*
- γ = Kelas data
- K = Fungsi kernel
- x_i = Data latih ke - i
- x^+ = Data latih kelas positif dengan nilai α tertinggi
- x^- = Data latih kelas negatif dengan nilai α tertinggi

6. Untuk pengklasifikasian data ke dalam kelas atau untuk melakukan identifikasi terhadap kelas dari data dapat diperoleh dengan menggunakan Persamaan. Apabila hasil dari $\text{sign}(h(x))$ adalah -1, dokumen termasuk dalam kelas bukan ujaran kebencian. Tetapi jika hasilnya +1, dokumen tersebut masuk ke dalam kelas ujaran kebencian.

$$h(x) = \sum_{i=0}^N \alpha_i \gamma_i K(x, x_i) + b \quad (2.14)$$

Keterangan:

- b = bias
- α = *Lagrange Multiplier*
- γ = Kelas data

- K = Fungsi kernel
- x_i = Data latih ke – i
- x = Data uji

2.7 Accuracy, Precision, Recall dan F-Measure

Accuracy atau akurasi merupakan sebuah ukuran evaluasi untuk membandingkan antara sebuah hasil keputusan diagnostik dengan *truth* atau kebenaran, akurasi sendiri merupakan sebuah bentuk pengukuran paling sederhana dari kualitas keputusan diagnostik yang merupakan nilai dari kasus di mana seorang psikiater benar dalam diagnosisnya (Metz, 1978). *Precision* dan *Recall* di definisikan sebagai kumpulan dokumen yang di *retrieve* dan kumpulan dokumen yang relevan (Perry, Kent, dan Berry, 1955). *Precision* mewakili bagian dokumen yang telah di *retrieve* yang relevan terhadap *query* dan dapat disebut sebagai *positive predictive value* yang kemudian dirumuskan sebagai:

$$Precision = \frac{|\{\text{jumlah data positif teridentifikasi benar}\} \cap \{\text{jumlah data positif}\}|}{|\{\text{jumlah data positif}\}|}$$

sedangkan *recall* mewakili bagian data yang bernilai positif yang berhasil diidentifikasi dan disebut *true positive rate* yang dapat di rumuskan sebagai:

$$Recall = \frac{|\{\text{jumlah data positif teridentifikasi benar}\} \cap \{\text{jumlah data teridentifikasi positif}\}|}{|\{\text{jumlah data teridentifikasi positif}\}|}$$

dan *f-measure* sebagai *harmonic means* atau rata-rata keseimbangan antara *precision* dan *recall* dari data

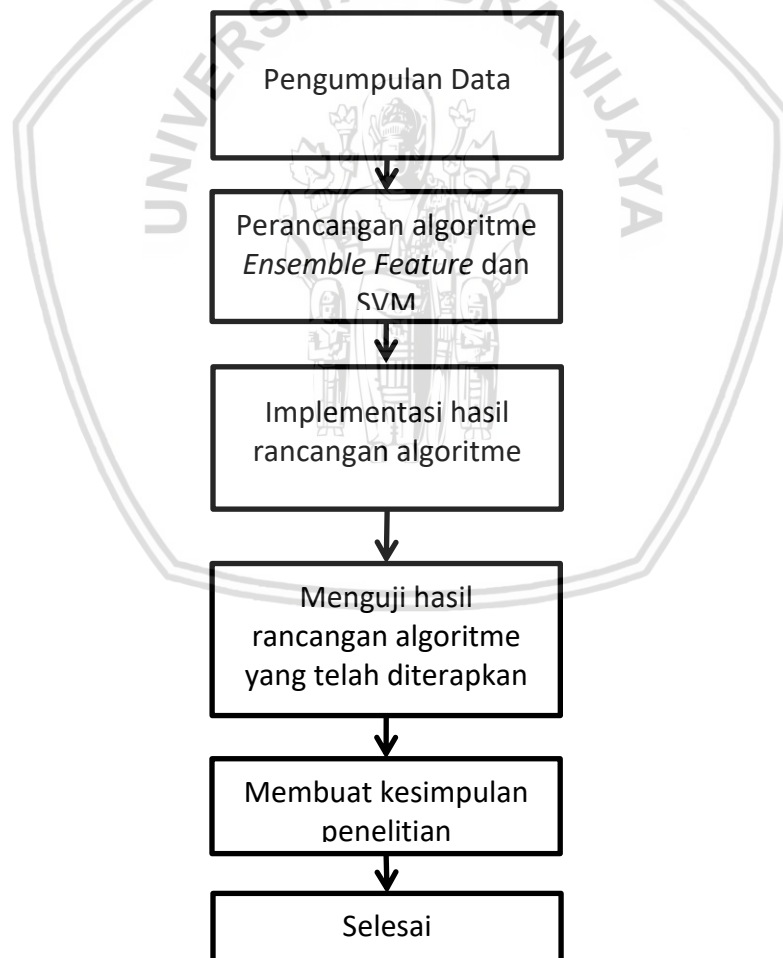
$$F\text{-Measure} = 2 \times \left(\frac{Precision \times Recall}{Precision + Recall} \right)$$

BAB 3 METODOLOGI

Bab metodologi berisi mengenai langkah-langkah atau tahapan dalam metodologi penelitian dan proses dalam melakukan penerapan algoritme *Ensemble Feature* dan *Support Vector Machine*(SVM) yang digunakan dalam penelitian. Dalam metode penelitian dijelaskan mengenai metode-metode yang digunakan dalam penelitian ini. Pada subbab perancangan, akan dijelaskan mengenai rancangan dalam proses menerapkan algoritme yang digunakan

3.1 Metodologi Penelitian

Subbab metodologi ini akan menjelaskan langkah yang akan ditempuh dalam proses penelitian. Penelitian ini dilakukan dengan beberapa tahap, yakni Pengumpulan Data, Perancangan, Implementasi, Pengujian, dan Pembuatan Kesimpulan.



Gambar 3.1 Tahapan Penelitian

3.2 Pengumpulan Data

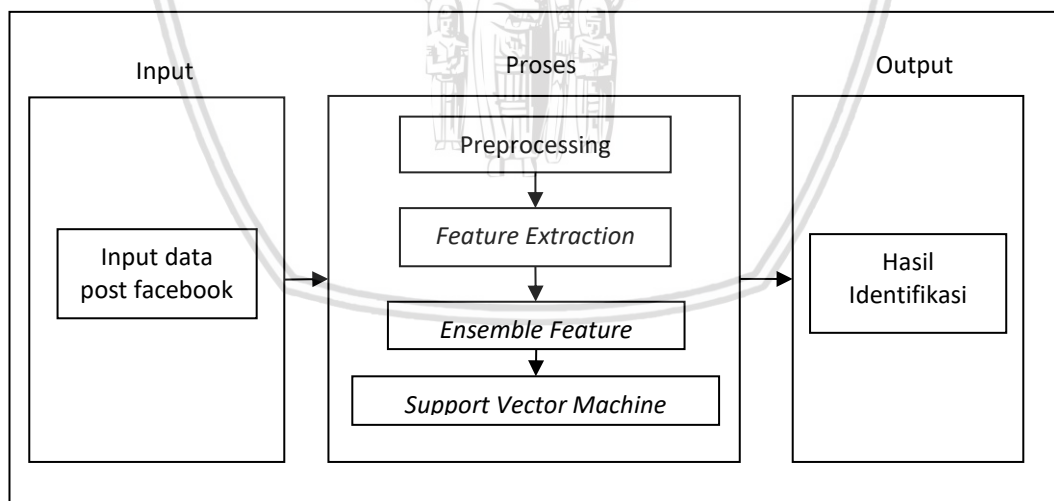
Penelitian ini dilakukan menggunakan data yang diambil langsung dari beberapa *public page* di *Facebook.com* melalui teknik *scraping* data yang terdiri dari 75 dataset berisi ujaran kebencian dan 75 dataset berisi bukan ujaran kebencian. Dataset berisikan huruf atau kalimat dari status seseorang di *Facebook* sehingga diperlukan proses preprosesing menggunakan pemrosesan teks, setelah preprosesing selesai dataset kemudian diekstraksi fiturnya dan dilakukan identifikasi oleh *Support Vector Machine*. Data-data ini kemudian dilabeli oleh seorang ahli bahasa untuk menentukan apakah data termasuk dalam ujaran kebencian atau bukan ujaran kebencian.

3.3 Perancangan

Tahapan perancangan sistem pada penelitian ini bertujuan untuk mempermudah proses dari implementasi dan pengujian. Perancangan sistem menghasilkan sebuah rancangan langkah kerja dari sistem yang sesuai dengan arsitektur sistem. Rancangan yang dihasilkan berupa model perancangan dan arsitektur dari sistem *Support Vector Machine*.

3.3.1 Model Perancangan

Model perancangan sistem bertujuan untuk menjelaskan proses kerja dari sistem secara terstruktur. Model ini akan mengilustrasikan proses sistem yang dimulai dari input yang dimasukkan sampai hasil yang dikeluarkan oleh sistem. Diagram model perancangannya adalah



Gambar 3.2 Model Perancangan Sistem

3.4 Implementasi

Tahapan implementasi merupakan tahapan yang dilakukan untuk membuat sistem berdasarkan perancangan yang telah dibuat. Selain itu, dalam

tahapan ini dilakukan proses implementasi terhadap beberapa informasi yang diperoleh dalam tahapan studi literature. Tahap yang dilakukan pada implementasi:

- Implementasi basis data untuk proses pengolahan dan penyimpanan data.
- Implementasi algoritme yang melakukan identifikasi dengan metode *Ensemble Feature* dan *Support Vector Machine* berbasis *Python*.
- Implementasi sistem ini akan menghasilkan identifikasi ujaran kebencian pada *Facebook*.

3.5 Pengujian

Pada penelitian ini akan digunakan beberapa jenis pengujian yaitu pengujian akurasi untuk menguji seberapa akurat algoritme yang digunakan, kemudian ada pengujian *precision* untuk menentukan kemampuan algoritme untuk me-*retrieve* dokumen yang relevan terhadap query dari seluruh dokumen yang di *retrieve* oleh algoritme, pengujian *recall* untuk menentukan kemampuan algoritme untuk secara sukses me-*retrieve* dokumen relevan, dan pengujian *f-measure* yang digunakan untuk mencari nilai rata-rata dari *precision*, dan *recall* dari algoritme yang digunakan.

3.6 Analisis

Tahapan analisis pada penelitian ini merupakan tahapan untuk melakukan analisis terhadap hasil identifikasi ujaran kebencian di *Facebook* dan akurasinya. Dari hasil aturan yang dapat mengidentifikasi ujaran kebencian di *Facebook* tersebut dapat dilakukan analisis terhadap metode *Ensemble Feature* dan *Support Vector Machine* yang digunakan untuk melakukan identifikasi. Semakin tinggi nilai akurasi dari hasil pengujian maka akan menunjukkan bahwa metode *Ensemble Feature* dan *Support Vector Machine* dapat menghasilkan nilai yang baik dalam melakukan identifikasi ujaran kebencian di *Facebook*. Hasil analisis ini kemudian akan digunakan untuk membuat kesimpulan dari penelitian ini.

3.7 Penarikan Kesimpulan

Tahapan penarikan kesimpulan pada penelitian ini dilakukan ketika seluruh tahapan perancangan, implementasi, dan pengujian metode yang digunakan telah selesai dilakukan. Kesimpulan kemudian akan diperoleh dari proses analisis terhadap metode yang digunakan. Setelah pembuatan kesimpulan kemudian akan dilakukan penulisan saran. Saran bertujuan untuk memperbaiki beberapa kesalahan yang terdapat dalam penelitian. Selain itu saran juga bertujuan untuk proses pengembangan penelitian selanjutnya.

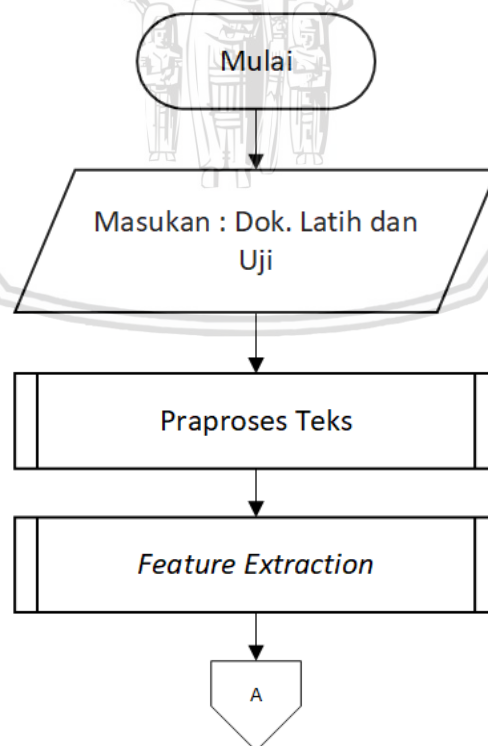
BAB 4 ANALISIS DAN PERANCANGAN SISTEM

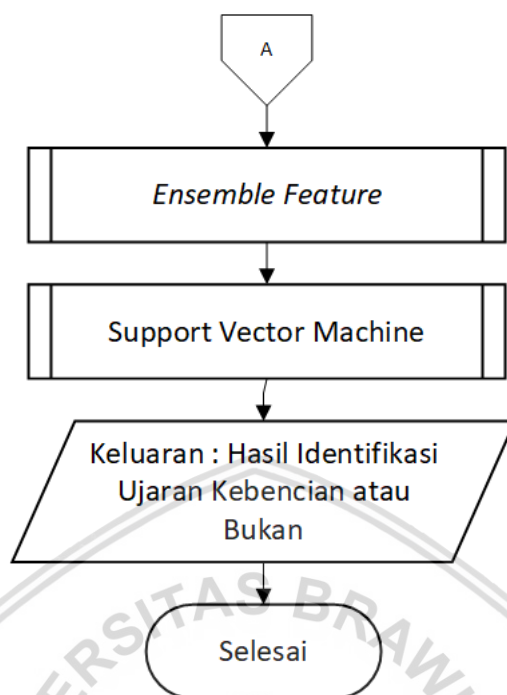
Pada bab ini akan diuraikan mengenai deskripsi permasalahan, deskripsi umum sistem, algoritme yang akan digunakan untuk menyelesaikan permasalahan dan perancangan proses pengujian.

4.1 Deskripsi Umum Sistem

Identifikasi ujaran kebencian pada *Facebook* dengan *Ensemble Feature* dan *Support Vector Machine* merupakan sistem yang dikembangkan untuk melakukan identifikasi terhadap *post* yang berisi ujaran kebencian pada media sosial *Facebook*. Data yang digunakan pada penelitian ini diambil dari beberapa halaman publik *Facebook* dengan menggunakan pengambilan data secara manual. Data yang telah diambil kemudian dibagi menjadi dua jenis data yaitu data latih dan data uji.

Data latih dan data uji tersebut kemudian akan dimasukkan ke dalam *pre-processing* teks untuk kemudian diolah pada perhitungan metode algoritme inti. Setelah tahap *pre-processing*, maka akan dilakukan *Feature Extraction* untuk mengambil fitur yang terdapat pada teks. Kemudian akan dibuat *Ensemble Feature* berdasarkan kebutuhan. Setelah itu akan dimasukkan ke dalam *Support Vector Machine* untuk mengetahui apakah data termasuk kelas positif atau negatif. Alur proses deskripsi umum ditunjukkan pada Gambar 4.1.

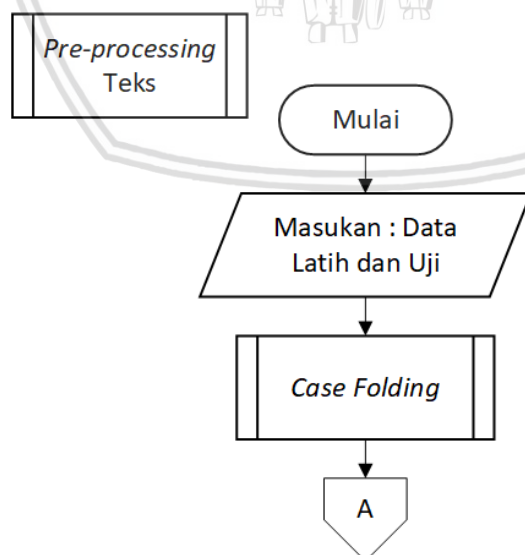


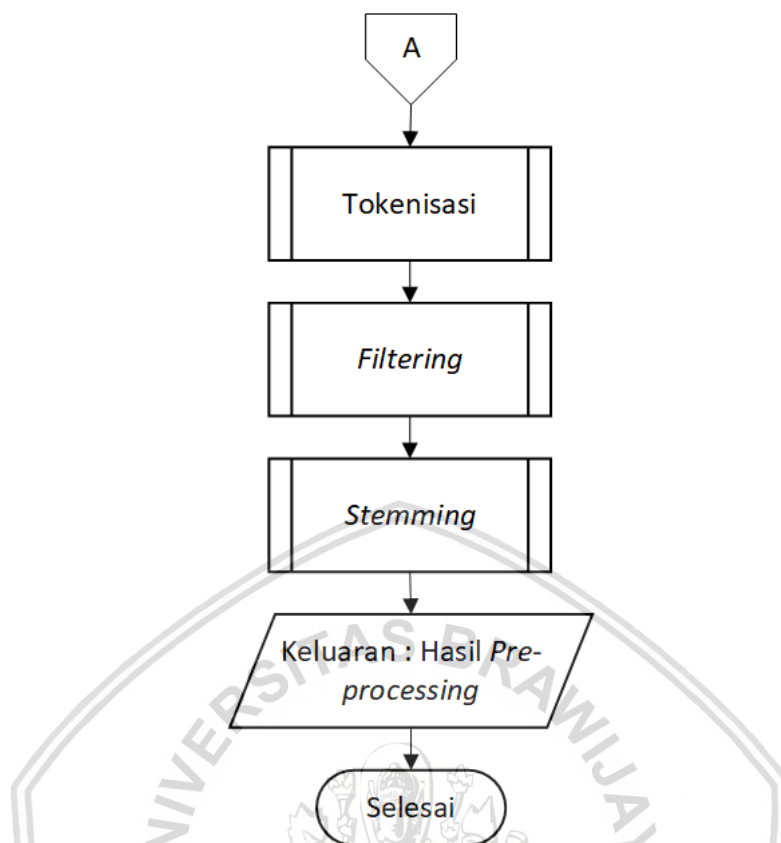


Gambar 4.1 Deskripsi Umum Sistem

4.2 Pre-processing teks

Pre-processing teks adalah langkah paling awal dalam tahapan pemrosesan teks dalam penelitian. Proses ini terdiri dari tokenisasi, *filtering*, dan *stemming*. Alur dari proses dalam tahapan *pre-processing* teks ini dapat dilihat pada Gambar 4.2.

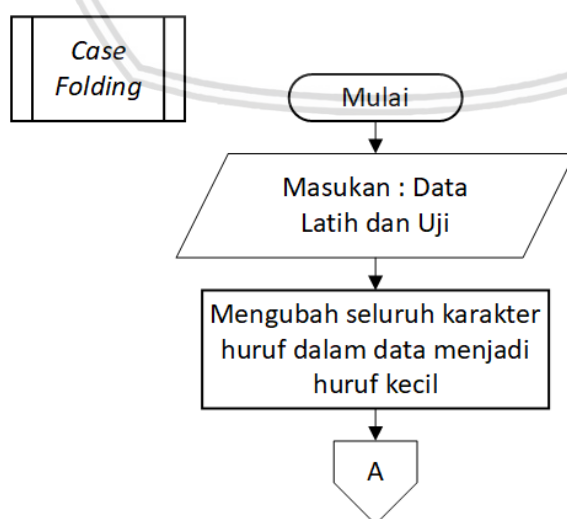


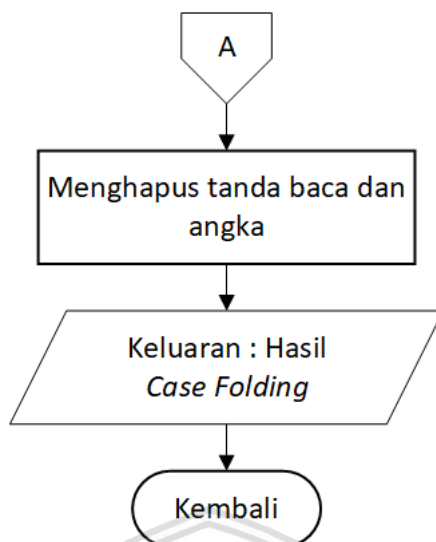


Gambar 4.2 Alur Proses *Pre-processing* teks

4.2.1 Case Folding

Case Folding adalah proses pertama dari tahapan *pre-processing* teks. Pada proses ini, teks yang berupa kalimat akan diubah menjadi kalimat berhuruf kecil seluruhnya. Alur proses *case folding* ditunjukkan pada Gambar 4.3.

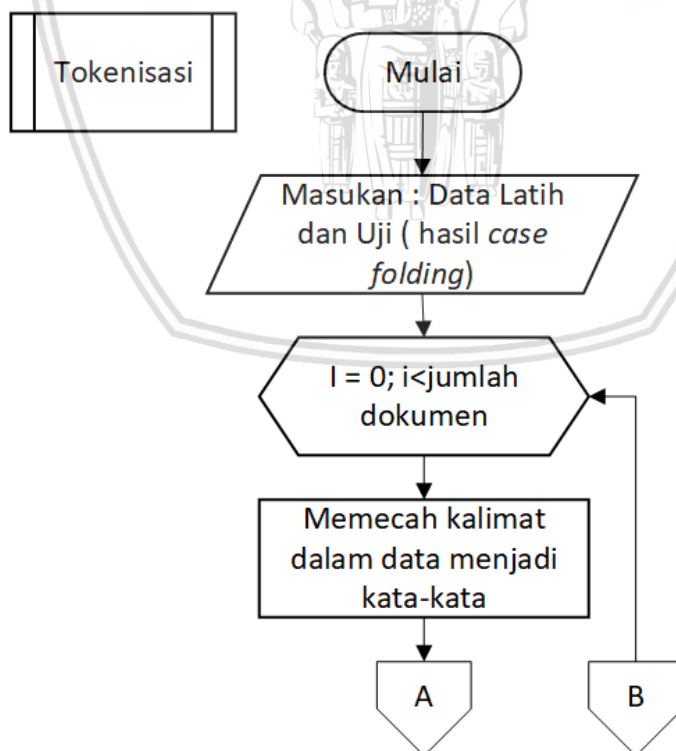


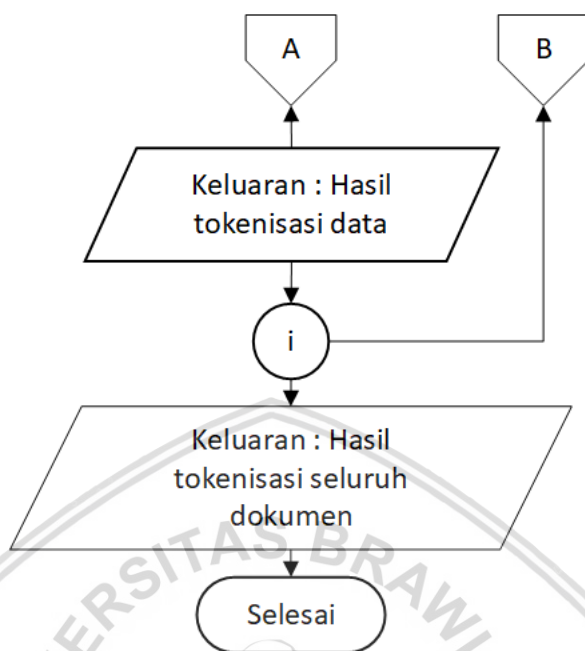


Gambar 4.3 Alur Proses *Case Folding*

4.2.2 Tokenisasi

Tokenisasi merupakan subproses dari tahapan *pre-processing* teks. Pada proses ini, teks yang pada awalnya berupa beberapa kalimat akan dipecah-pecah menjadi tiap kata atau token. Alur proses tokenisasi ditunjukkan pada Gambar 4.4.

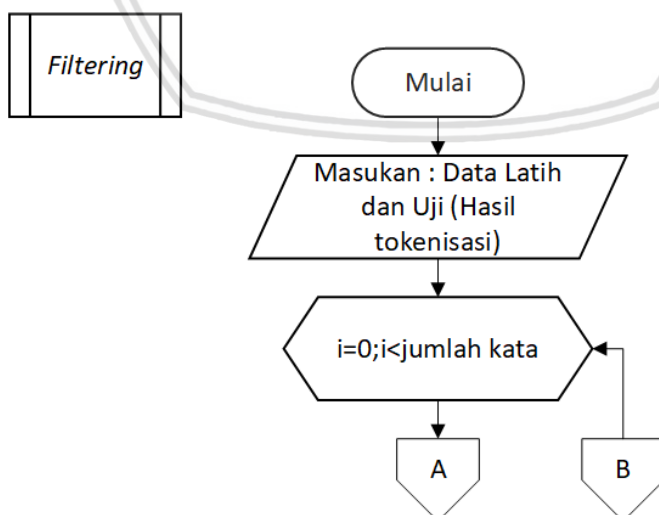


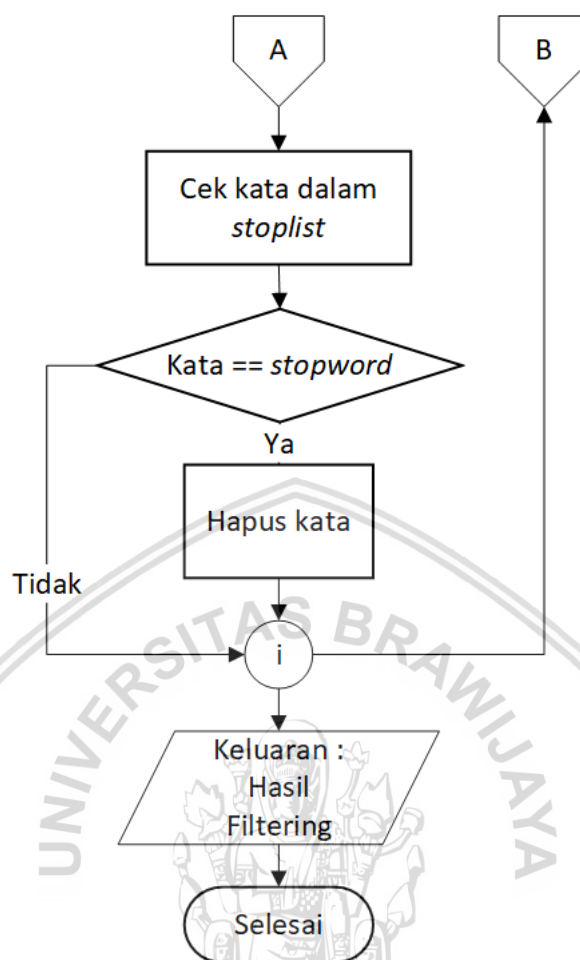


Gambar 4.4 Alur Proses Tokenisasi

4.2.3 Filtering

Filtering juga merupakan subproses dari *pre-processing* teks yang memiliki fungsi untuk menghapus kata yang tidak memiliki makna khusus dan tidak berpengaruh untuk hasil akhir identifikasi (*stopword*). Alur proses *Filtering* ditunjukkan pada Gambar 4.5

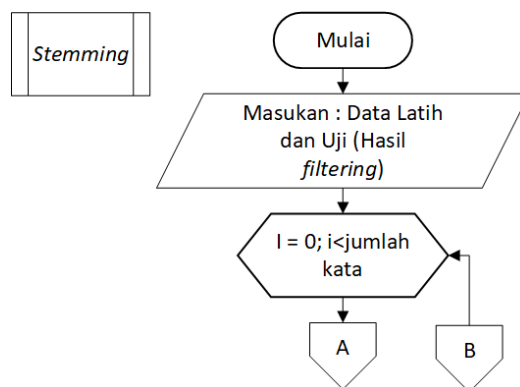


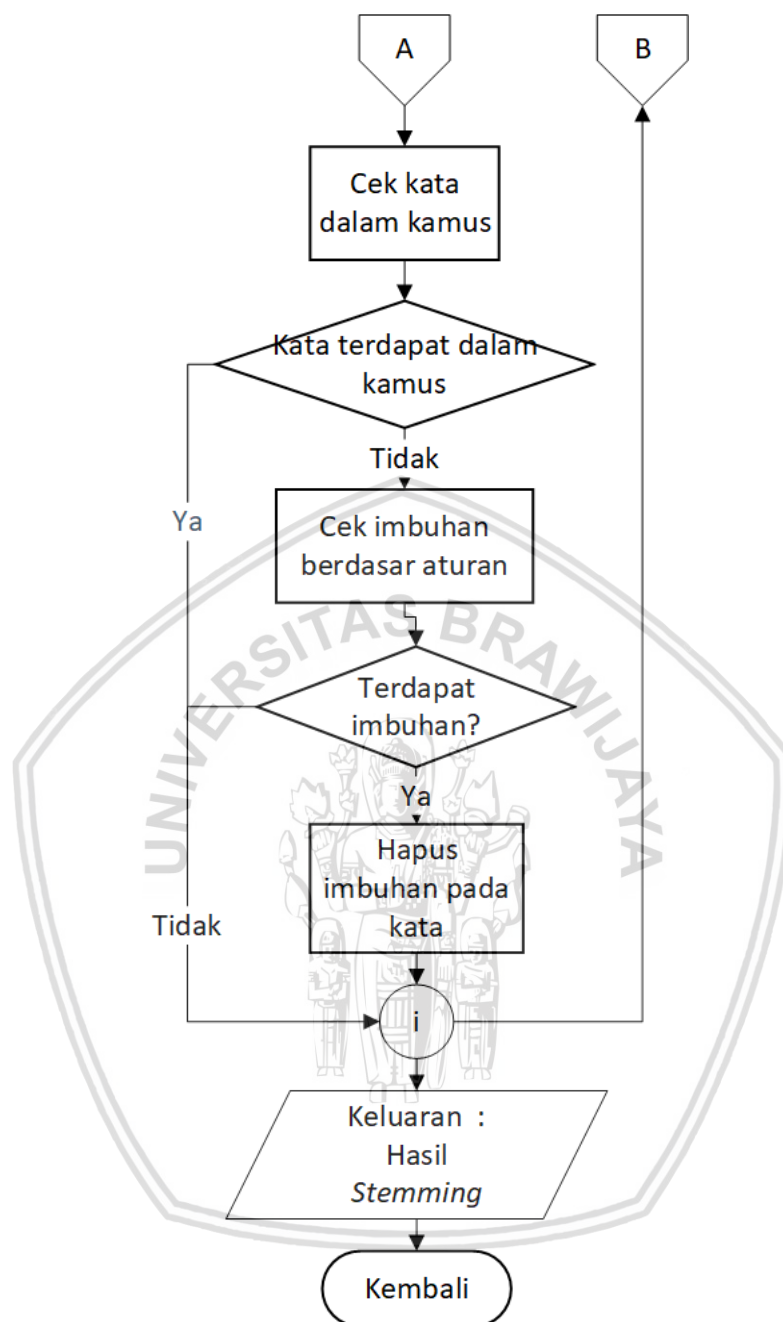


Gambar 4.5 Alur Proses *Filtering*

4.2.4 *Stemming*

Stemming adalah proses paling akhir dari tahapan *pre-processing* teks. Pada proses *stemming*, dokumen hasil *filtering* yang berupa kata yang memiliki imbuhan akan diubah menjadi kata dasar dengan cara menghapus imbuhan. Alur proses *stemming* ditunjukkan pada Gambar 4.6





Gambar 4.6 Alur Proses Stemming

4.3 Feature Extraction

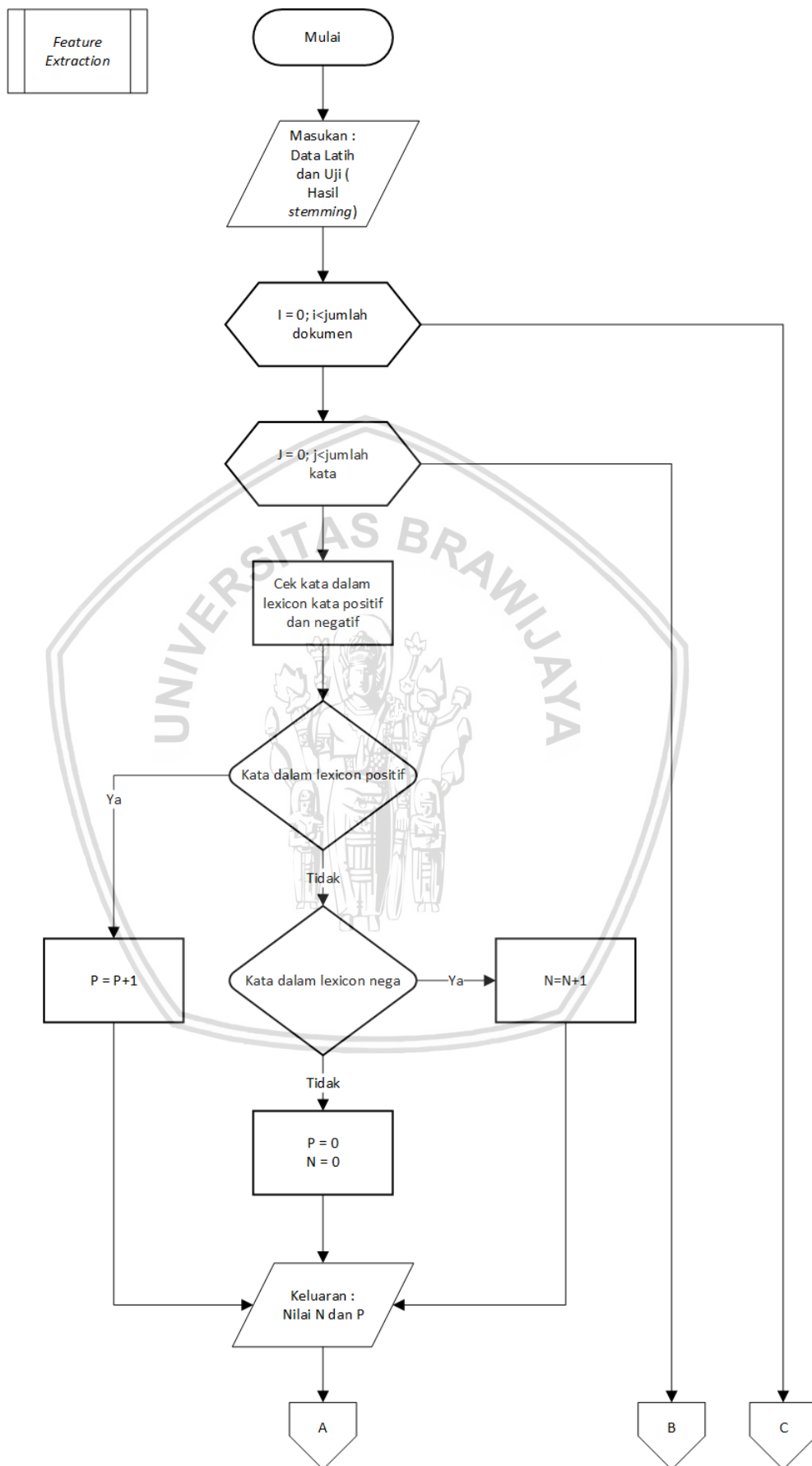
Feature Extraction merupakan proses untuk mendapatkan fitur-fitur dari teks yang dimasukkan. Fitur yang dimaksud adalah fitur-fitur yang akan digunakan

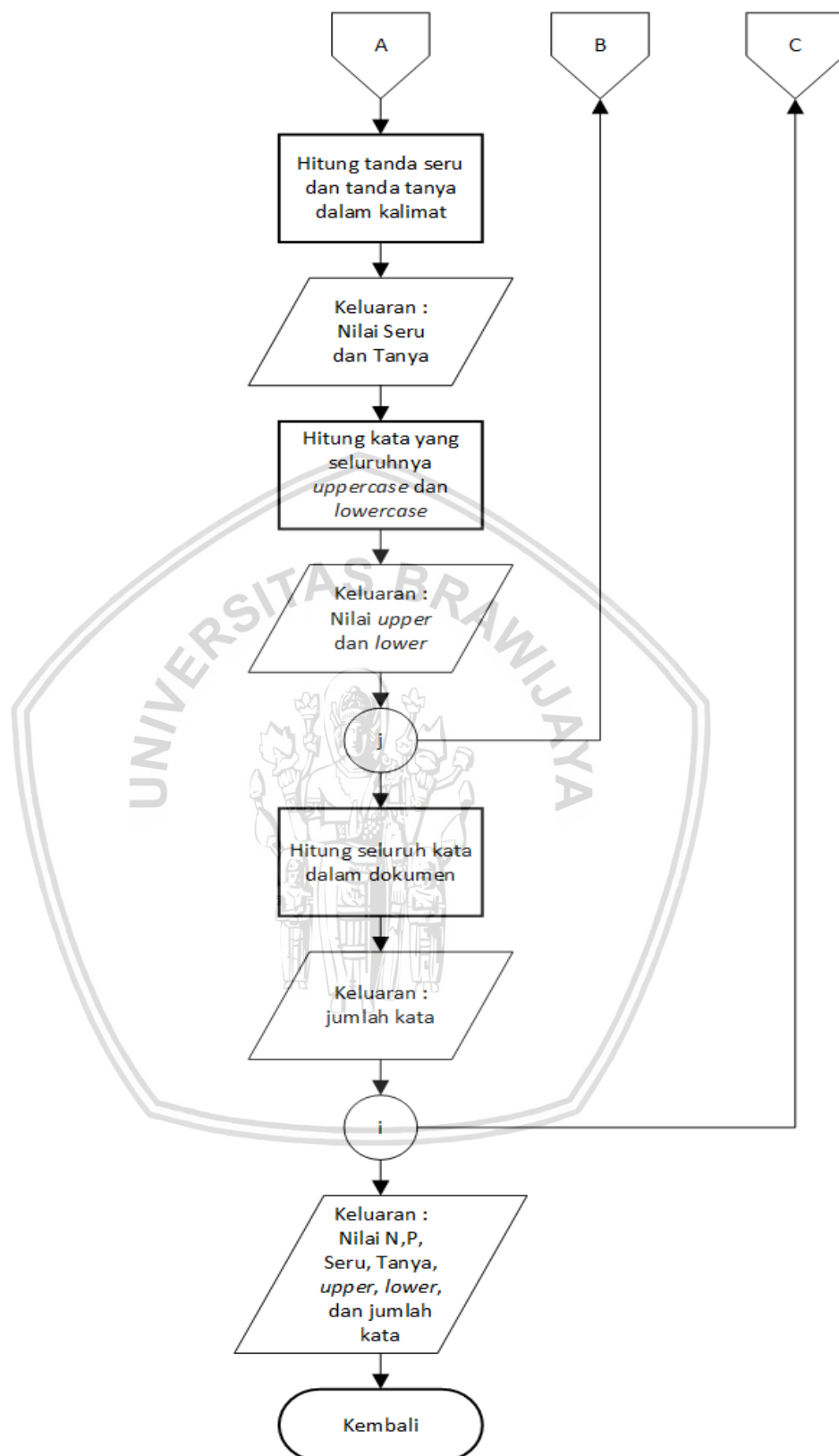
untuk proses *Ensemble Feature*. Fitur-fitur yang dimaksud berada pada Tabel 4.1. Alur dari proses *Feature Extraction* ditunjukkan pada Gambar 4.7.

Tabel 4.1 Daftar Fitur

Kategori Fitur	Fitur	Keterangan Fitur
Fitur khusus terkait Facebook	F1	Jumlah <i>like react</i> pada post
	F2	Jumlah <i>angry react</i> pada post
	F3	Jumlah komentar pada post
	F4	Jumlah <i>share post</i>
Fitur Tekstual	F5	Jumlah kata dalam post
	F6	Jumlah tanda seru dalam post
	F7	Jumlah tanda tanya dalam post
	F8	Jumlah kata dengan huruf capital
	F9	Jumlah kata dengan huruf kecil
Fitur Terkait Lexicon	F10	Kata berkonotasi Negatif
	F11	Kata berkonotasi Positif

Untuk daftar kata berkonotasi negatif dan berkonotasi positif digunakan daftar kata positif yang berasal dari penelitian yang dilakukan oleh Wahid, D.H., & Azhari, S.N. pada tahun 2016 yang meneliti mengenai peringkasan sentiment ekstraktif di Twitter menggunakan hybrid *TF-IDF* dan *Cosine Similarity* dengan menggunakan daftar kata yang serupa, dan dari penelitian yang dilakukan oleh Liu, Bing, Hu, Minqing, & Cheng, Junsheng pada tahun 2005 yang meneliti tentang cara menganalisa dan membandingkan opini pada *world wide web* secara otomatis.

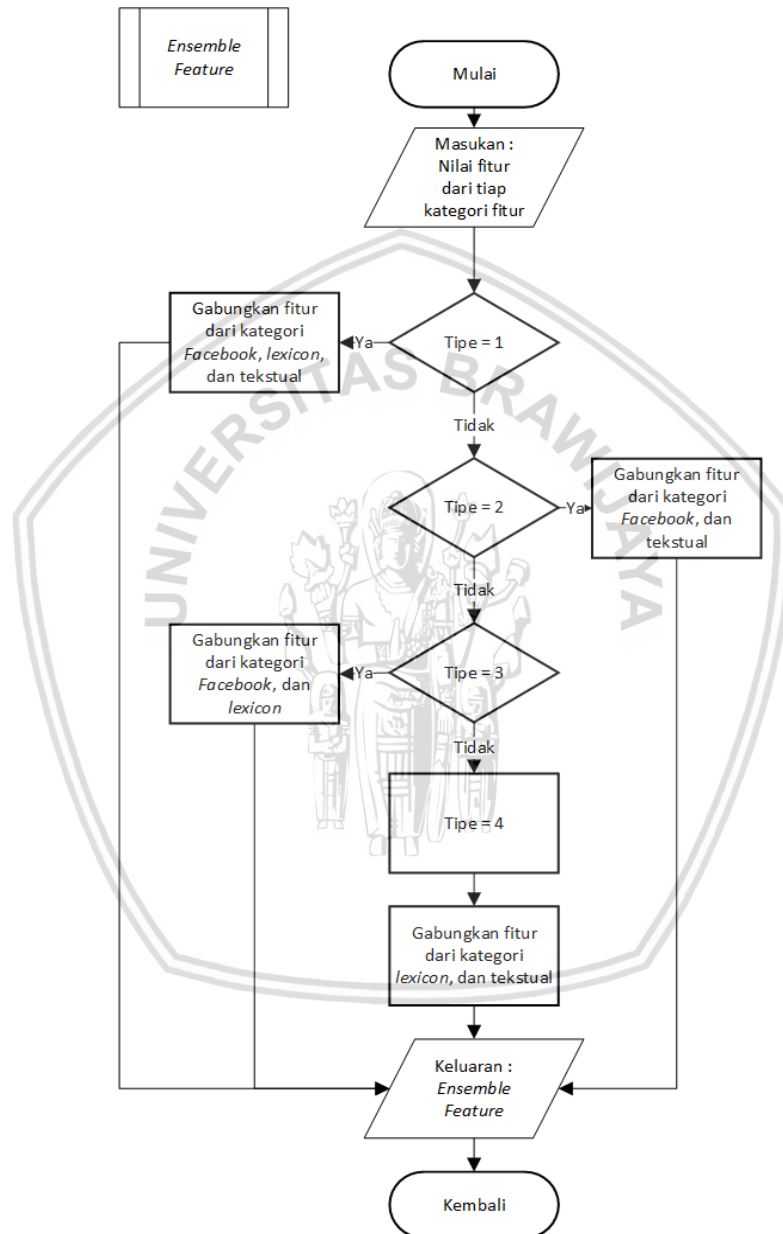




Gambar 4.7 Alur Proses *Feature Extraction*

4.4 Ensemble Feature

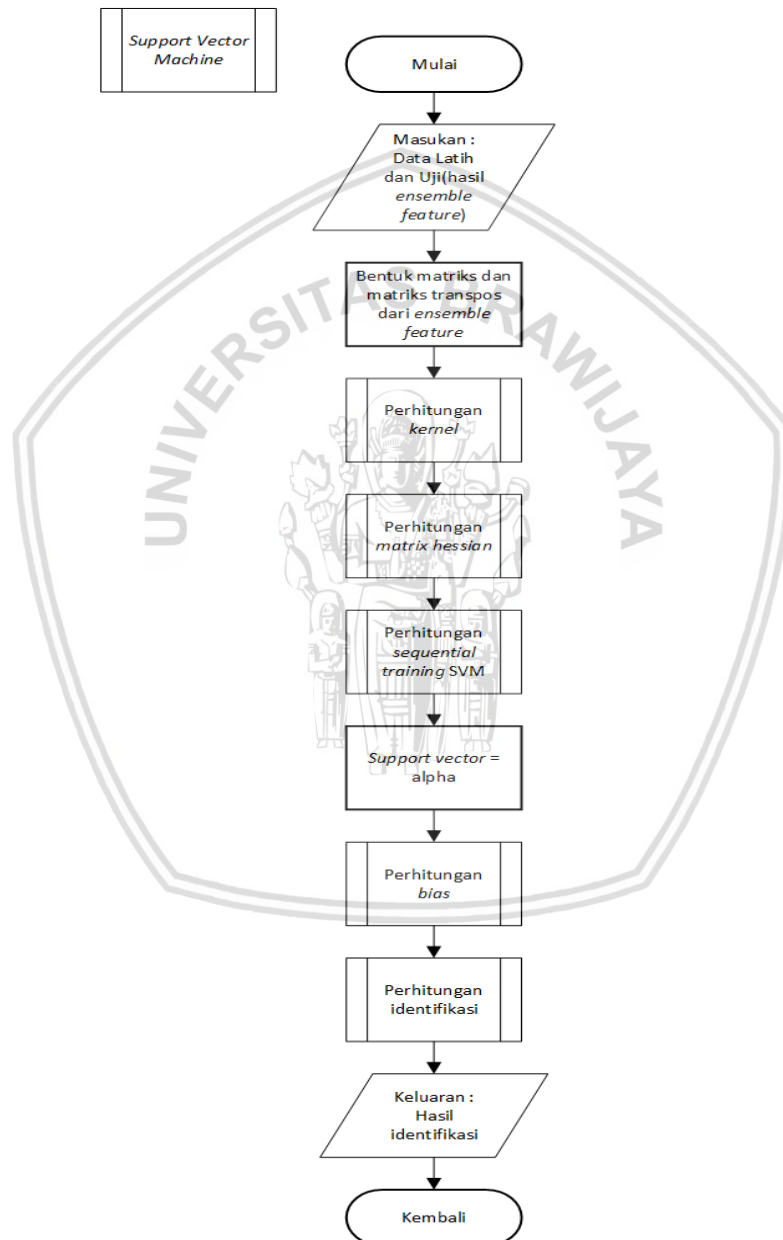
Ensemble Feature merupakan proses penggabungan beberapa kategori fitur menjadi satu. Untuk penggabungan fitur dilakukan sesuai kebutuhan yaitu dengan menggabungkan seluruh kategori fitur menjadi satu atau menggabungkan beberapa kategori fitur saja. Alur proses dari metode *Ensemble Feature* ditunjukkan pada Gambar 4.8



Gambar 4.8 Alur Proses *Ensemble Feature*

4.5 Support Vector Machine

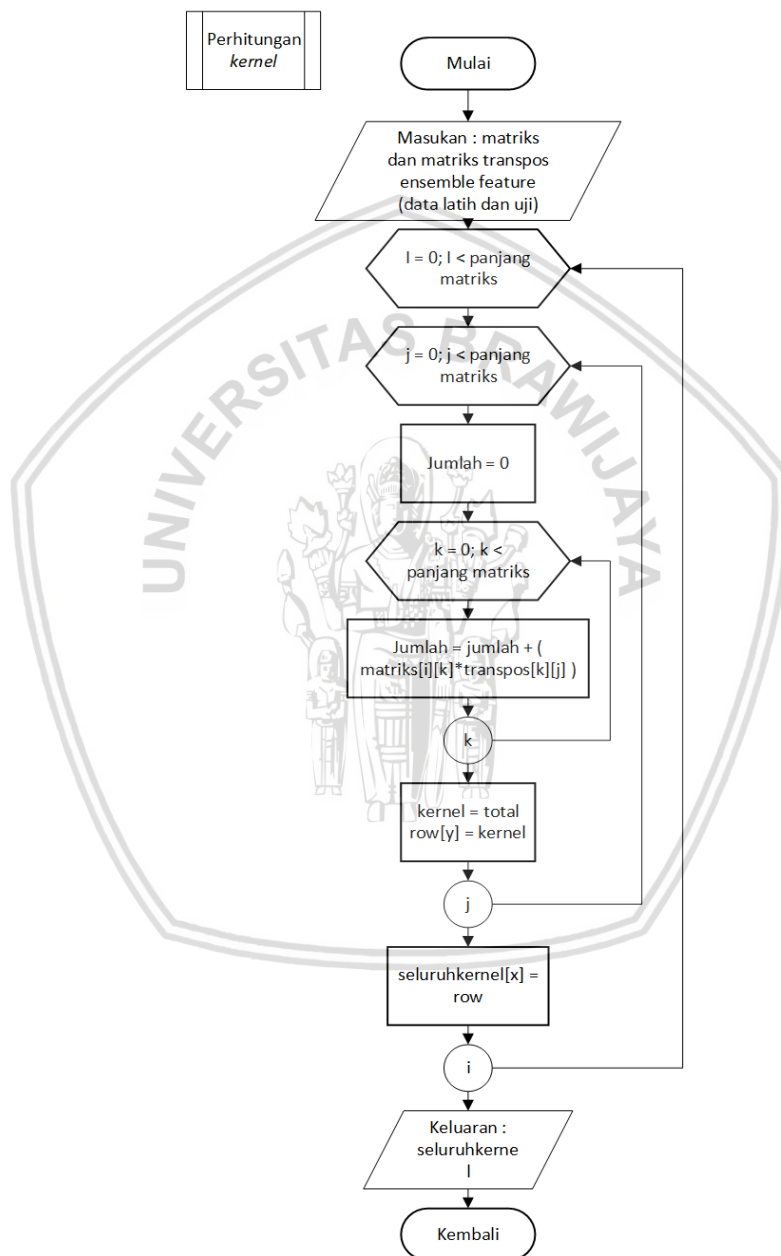
Support Vector Machine adalah algoritme utama dalam sistem identifikasi ujaran kebencian ini. Metode ini akan melakukan pembagian terhadap dokumen teks yang telah melalui beberapa proses sebelumnya seperti *pre-processing* teks, *feature extraction*, dan *ensemble feature* menjadi dua jenis kelas yaitu kelas ujaran kebencian (positif) dan kelas bukan ujaran kebencian (negatif) berdasarkan nilai atau bobot dari tiap fitur dalam kalimat tersebut. Alur proses dari metode *Support Vector Machine* ditunjukkan pada Gambar 4.9



Gambar 4.9 Alur Proses *Support Vector Machine*

4.5.1 Perhitungan Kernel

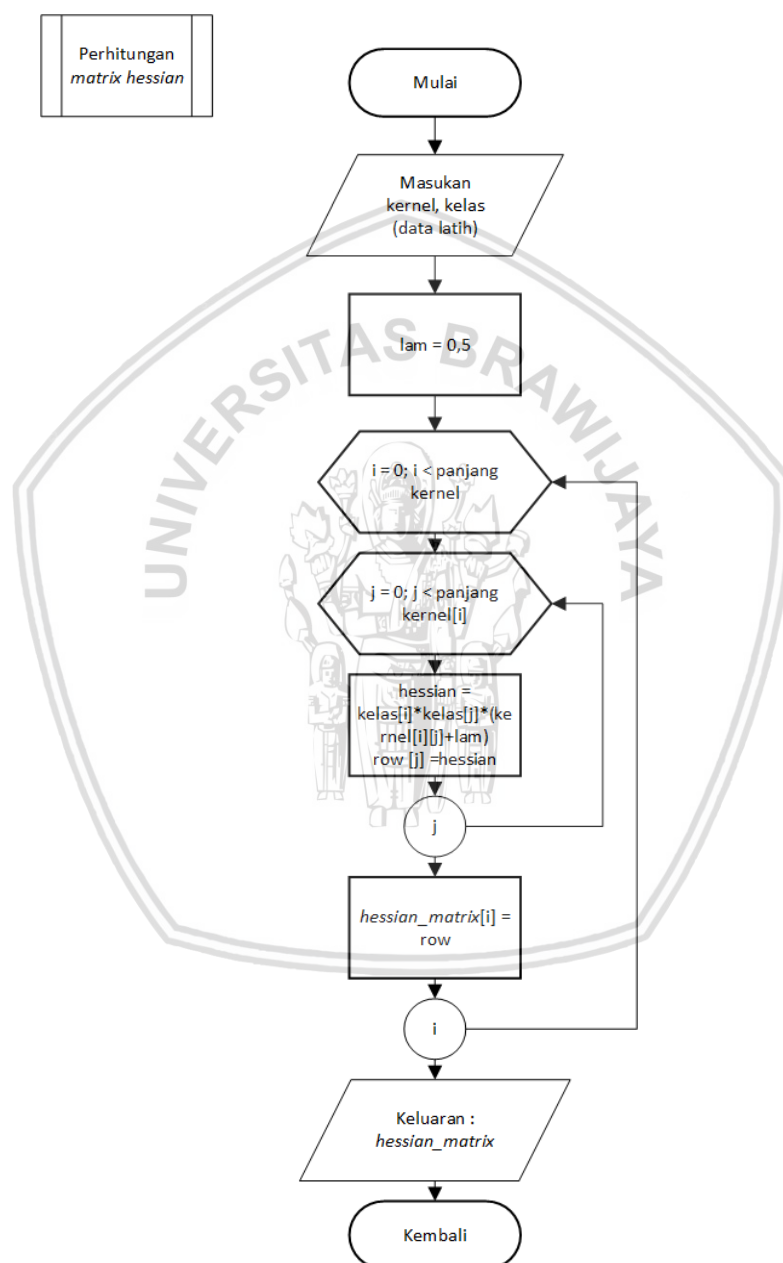
Perhitungan kernel adalah tahap awal dalam proses metode *Support Vector Machine*. Dalam tahapan ini, akan dilakukan perkalian antara matriks dari proses *Ensemble Feature* dengan matriks transpos dari data latih setelah itu hasil dari perkaliannya akan dimasukkan ke dalam sebuah rumus untuk proses perhitungan dengan jenis kernel *linear*. Alur proses dari tahap perhitungan kernel ditunjukkan pada Gambar 4.10



Gambar 4.10 Alur Proses Perhitungan Kernel

4.5.2 Perhitungan Matriks Hessian

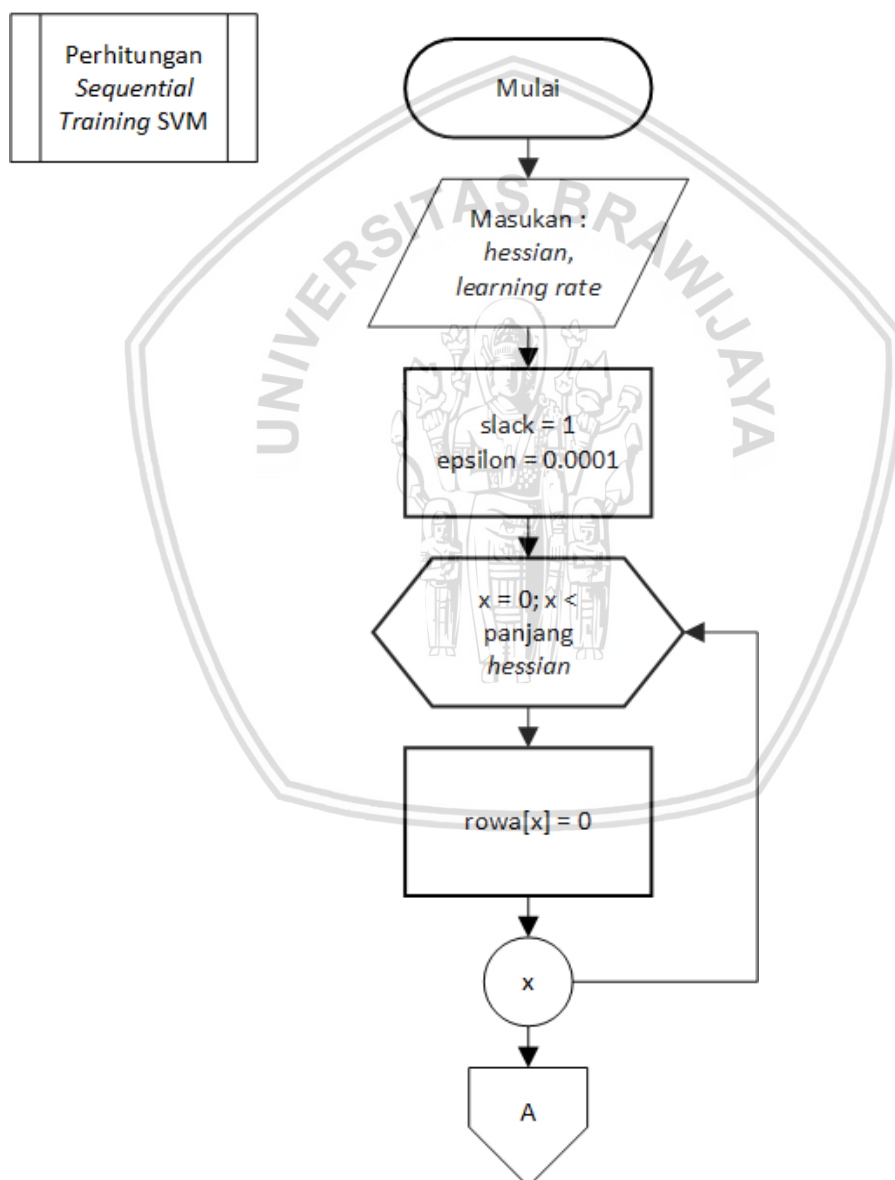
Perhitungan matriks hessian adalah langkah berikutnya dalam proses perhitungan metode *Support Vector Machine*. Dalam proses ini, hasil dari proses perhitungan kernel dari data ke-x dan ke-y akan dimasukkan ke dalam sebuah Persamaan matriks hessian dan dikali dengan kelas ke-x dan ke-y dengan kelas positif bernilai 1 dan kelas negatif bernilai -1. Alur proses perhitungan matriks hessian ditunjukkan pada Gambar 4.11

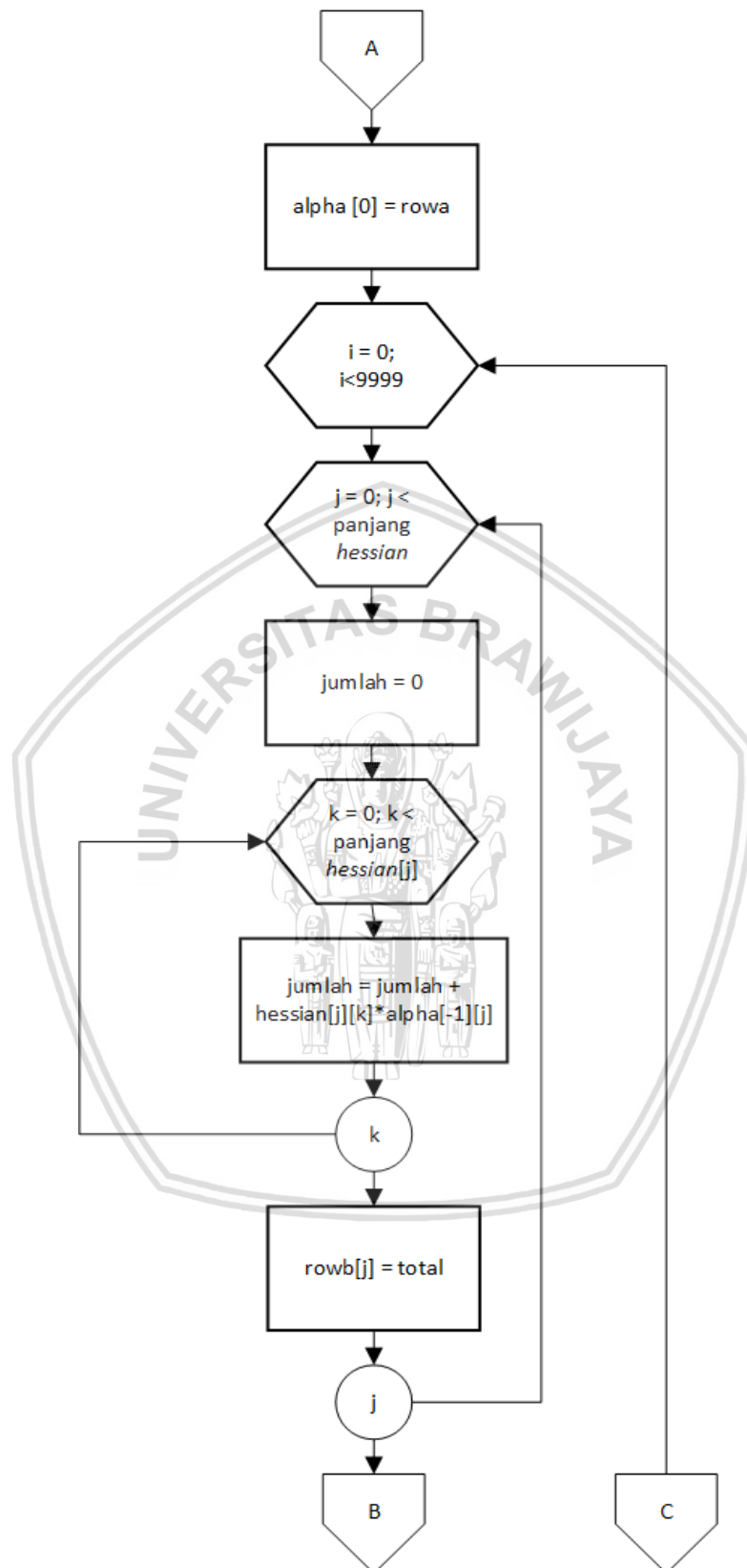


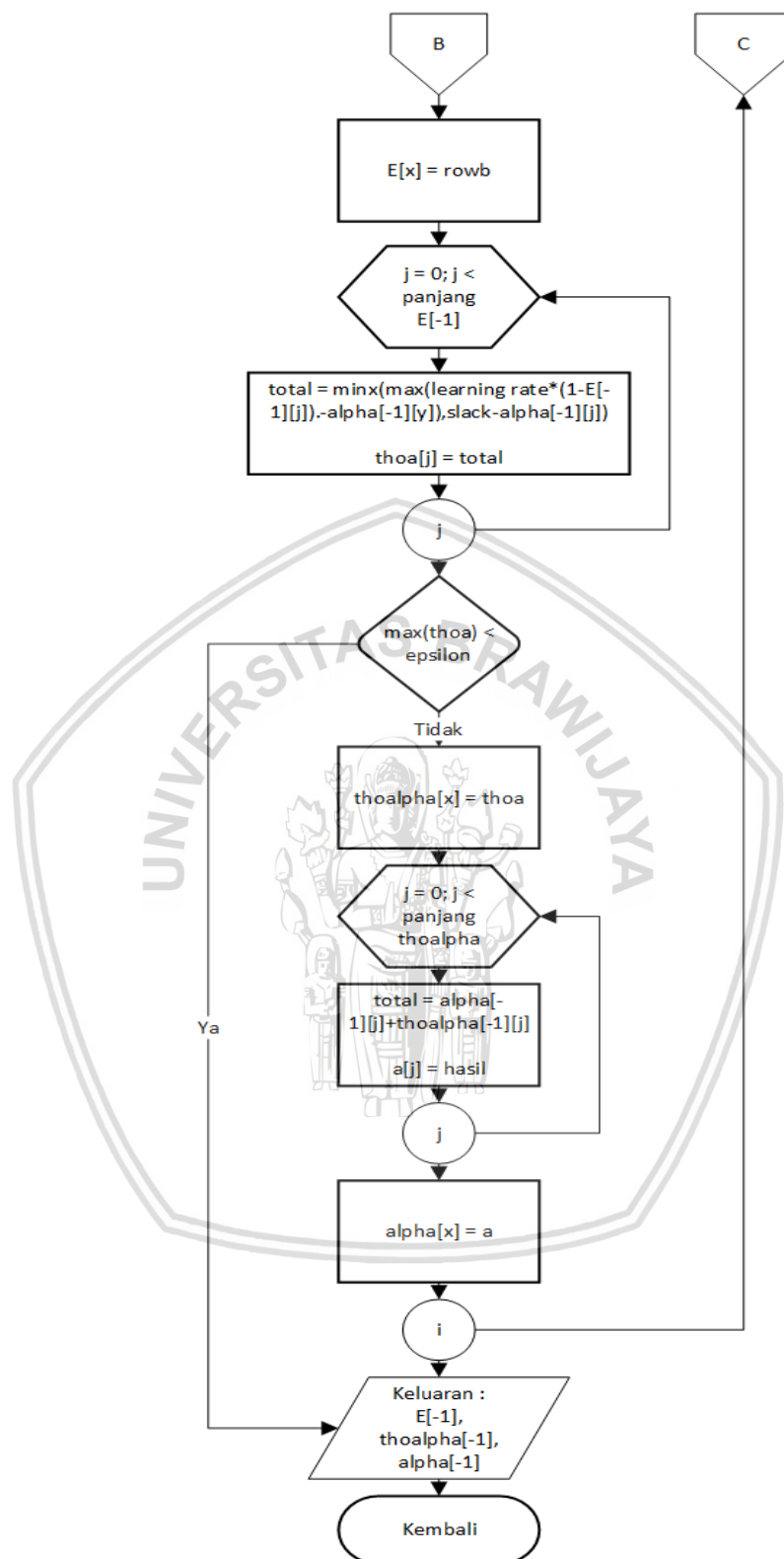
Gambar 4.11 Alur Proses Perhitungan Matrix Hessian

4.5.3 Perhitungan *Sequential Training SVM*

Perhitungan *sequential training SVM* adalah proses berikutnya dalam proses perhitungan metode *Support Vector Machine*. Pada tahap ini, terdapat perhitungan nilai error(E). Apabila nilai maksimal dari perubahan alpha lebih kecil dari nilai epsilon, iterasi akan berhenti. Nilai yang dihasilkan sebelum iterasi berhenti akan dijadikan sebagai nilai *support vector* atau titik data paling dekat dengan *hyperplane* yang merupakan garis pembatas antar kelas data. Alur proses dari perhitungan *sequential training SVM* ditunjukkan pada Gambar 4.12



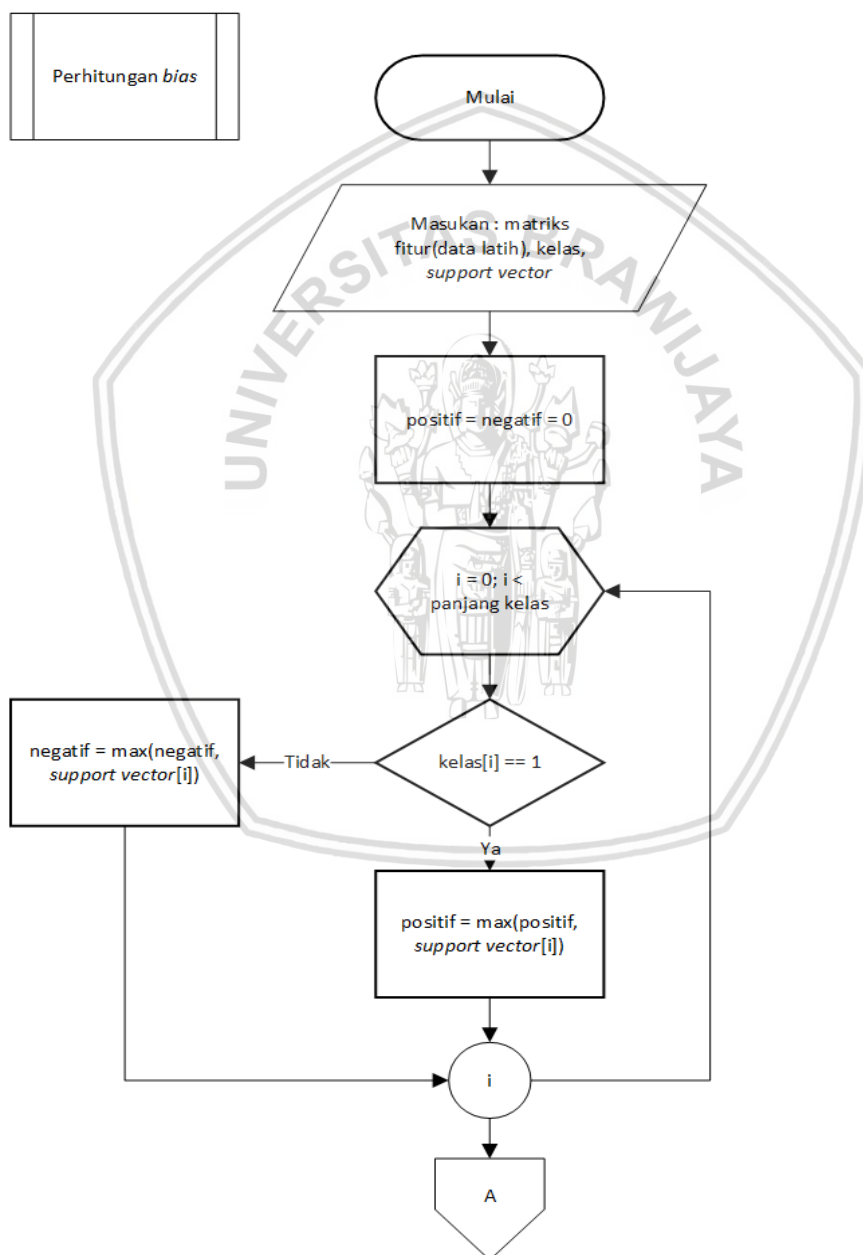


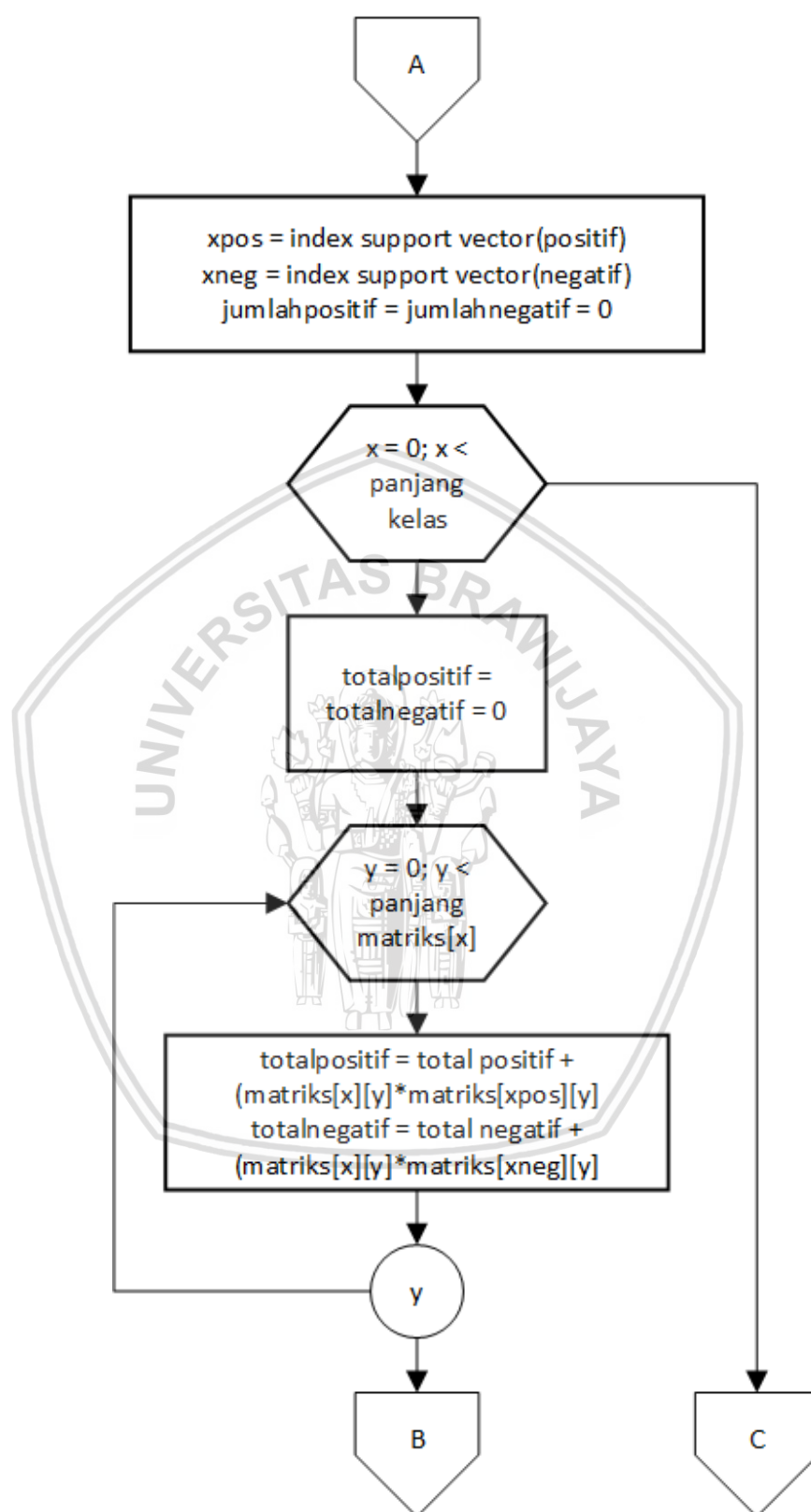


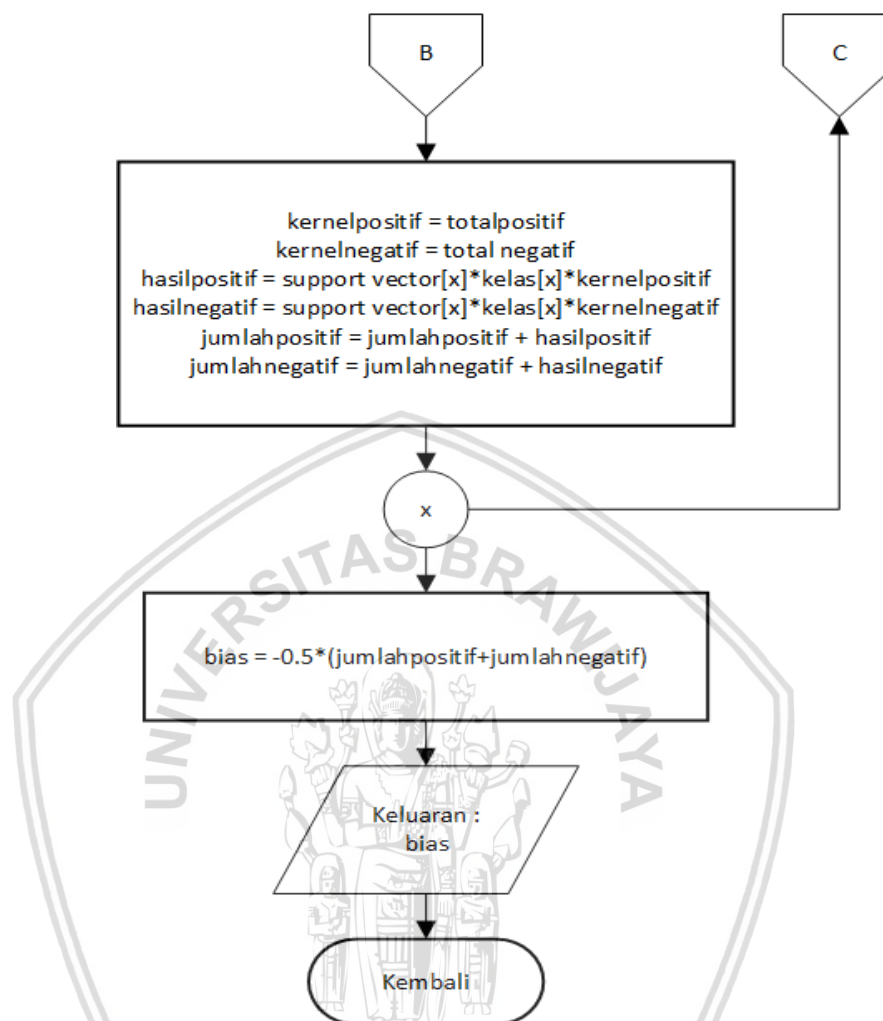
Gambar 4.12 Alur Proses Perhitungan *Sequential Training SVM*

4.5.4 Perhitungan *Bias*

Perhitungan *bias* merupakan tahapan sebelum tahap terakhir dalam proses perhitungan *Support Vector Machine* yang digunakan untuk membentuk sebuah *hyperplane*. Dalam tahap ini, dari *support vector* yang terpilih akan dipilih sebuah *support vector* yang memiliki nilai a tertinggi dari masing-masing kelas sebagai sebuah pembatas antar tiap kelas data. Kemudian dari beberapa *support vector* tersebut akan dilakukan perhitungan kernel *linear*-nya dan dimasukkan ke dalam perhitungan *bias*. Alur proses dari perhitungan *bias* ditunjukkan pada Gambar 4.12



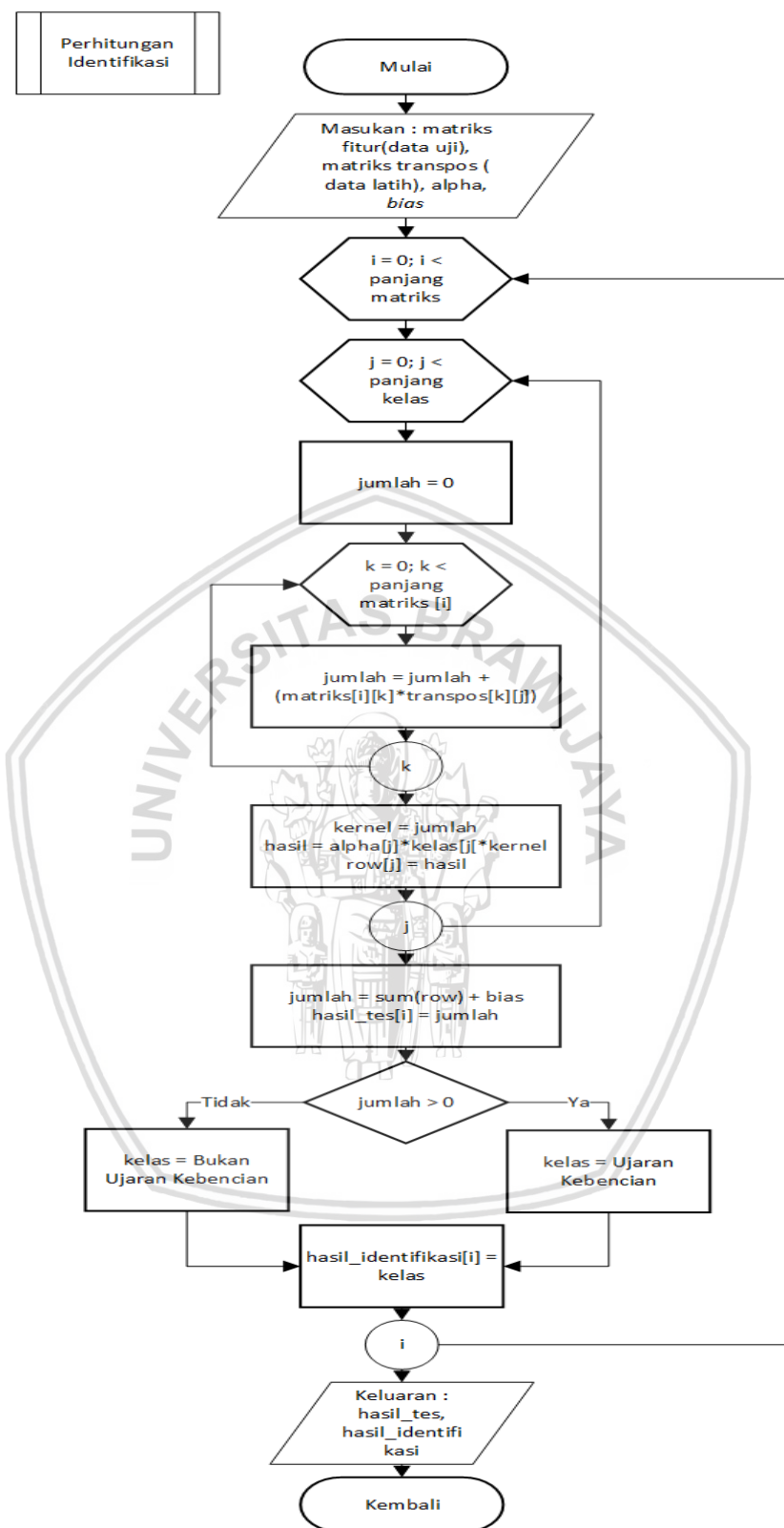




Gambar 4.13 Alur Proses Perhitungan *Bias*

4.5.5 Perhitungan Identifikasi

Perhitungan identifikasi adalah tahap terakhir dalam proses perhitungan metode *Support Vector Machine* untuk melakukan identifikasi terhadap data uji. Akan dilakukan perkalian antara data latih dan data uji untuk mendapatkan kernel *linear*, dan kemudian dimasukkan ke dalam Persamaan fungsi *hyperplane* yang kemudian dikalikan dengan nilai α dan kelas berdasarkan data latih yang kemudian ditambahkan nilai *bias*. Jika hasilnya lebih dari 0, akan masuk ke dalam kelas ujaran kebencian. Namun jika terjadi sebaliknya maka akan masuk ke dalam kelas bukan ujaran kebencian. Alur proses dari metode perhitungan identifikasi ditunjukkan pada Gambar 4.14



Gambar 4.14 Alur Proses Perhitungan Identifikasi

4.6 Manualisasi Metode *Ensemble Feature* dan *Support Vector Machine*

Pada tahap ini, dokumen teks yang berupa *post-post* dari laman publik akan diproses dengan menggunakan proses perhitungan dari metode *Support Vector Machine*. Dokumen teks diambil menggunakan *scraping* data dan terbagi menjadi data uji dan data latih. Data latih adalah kumpulan dokumen teks yang telah diketahui kelas dari tiap datanya yaitu ujaran kebencian dan bukan ujaran kebencian. Sedangkan data uji sendiri merupakan dokumen teks yang belum diketahui kelasnya dan akan diproses dengan metode *Support Vector Machine* untuk mengetahui kelas datanya. Sampel atau contoh untuk masing-masing data uji dan data latih dapat dilihat pada Tabel 4.2 dan 4.3

Tabel 4.2 Sampel Data Latih

No.	Data	Kelas (y)
1	Semuanya difitnah Penebar SARA dengan ujaran kebencian Padahal Semua Anti Kezaliman & Berjasa Besar dalam MELENGSERKAN & MELONGSORKAN AHOK SI PENISTA AGAMA	1
2	Bagi Cecunguk Koh Napi, Menebar Fitnah Setiap Hari Hukumnya Adalah WAJIB!	1
3	PILKADA JAKARTA BUKAN SEKEDAR PERTARUNGAN ANTARA ANIS vs AHOK	1
4	Hal Biasa Tokoh Masyarakat Menemui Habib Rizieq, Karena Kalau Yang Mereka Temui Si DeSi Atau Abu Janda Bisa APESSS Dan KALAH SUPER TELAK Lagi Seperti Ahok- Djarot Kemarin	1
5	Cecunguk Owe Makin Blo'on Aja. Udah Tau Owe Kalah Telak Selisih Hampir Satu Juta Suara Masih Aja Mereka Ngaku Mayoritas Dan Ngajakin Ngirim Karangan Bunga Sampah Begini.	1
6	Jika membahas PKI dituduh sebagai upaya untuk menyerang pemerintah, maka tanpa sadar para IQ 200 Sekolam ini berpendapat bahwa pemerintah RI itu PKI.	-1
7	Jika PKI sudah tidak ada lagi di Indonesia,	1

	lantas kenapa kok banyak yang blingsatan ketika kita mau memutar film G30SPKI ???	
8	Mungkin kalian sudah demikian takutnya kehilangan kekuasaan ya? Kok segala sesuatu dihubung-hubungkan dengan politik?	-1
9	Jika ada orang yang demikian gampangya menuduh orang lain sebagai pecandu PCC dan FLAKKA, maka mungkin dia... Ah.... Tak tega aku meneruskannya bang!	1
10	Hoax dan kebencian dari kubu mereka dipelihara, sementara dari pihak kita... ya gitu deh.	1

Tabel 4.3 Sampel Data Uji

No.	Data	Kelas (y)
1	MEDIA BAJINGAN & PENJAHAT SOSMED	-
2	Penundaan Sidang Penista Agama dengan agenda pembacaan Rencana Tuntutan oleh JPU dengan alasan belum siap karena belum selesai diketik adalah TRAGEDI PENEGAKAN HUKUM.	-
3	Ahok untuk Kaum Zhongers alias pendukungnya saja dan mengayomi hanya kelompok pendukungnya saja.	-

4.6.1 Case Folding

Tahap pertama dalam manualisasi ini adalah melakukan *case folding* pada data, yaitu proses untuk mengubah data berupa kalimat dengan huruf capital menjadi huruf kecil seluruhnya. Manualisasi *case folding* pada data dapat dilihat pada Tabel 4.4 dan 4.5

Tabel 4.4 Data Latih yang mengalami proses *case folding*

No.	Data asli	Setelah proses case folding
1	Semuanya difitnah Penebar SARA dengan ujaran kebencian Padahal Semua Anti Kezaliman	'semuanya difitnah penebar sara dengan ujaran kebencian padahal semua anti kezaliman & berjasa besar dalam melengserkan & melongsorkan ahok si penista

	& Berjasa Besar dalam MELENGSERKAN & MELONGSORKAN AHOK SI PENISTA AGAMA	agama'
2	Bagi Cecunguk Koh Napi, Menebar Fitnah Setiap Hari Hukumnya Adalah WAJIB!	'bagi cecunguk koh napi, menebar fitnah setiap hari hukumnya adalah wajib!'
3	TAPI PERTARUNGAN ANTARA: PEMBELA ISLAM vs PENISTA ISLAM	'tapi pertarungan antara pembela islam vs penista islam'
4	Hal Biasa Tokoh Masyarakat Menemui Habib Rizieq, Karena Kalau Yang Mereka Temui Si DeSi Atau Abu Janda Bisa APESSS Dan KALAH SUPER TELAK Lagi Seperti Ahok-Djarot Kemarin	'hal biasa tokoh masyarakat menemui habib rizieq, karena kalau yang mereka temui si desi atau abu janda bisa apesss dan kalah super telak lagi seperti ahok- djarot kemarin'
5	Cecunguk Owe Makin Blo'on Aja. Udah Tau Owe Kalah Telak Selisih Hampir Satu Juta Suara Masih Aja Mereka Ngaku Mayoritas Dan Ngajakin Ngirim Karangan Bunga Sampah Begini.	'cecunguk owe makin blo'on aja. udah tau owe kalah telak selisih hampir satu juta suara masih aja mereka ngaku mayoritas dan ngajakin ngirim karangan bunga sampah begini. '
6	Jika membahas PKI dituduh sebagai upaya untuk menyerang pemerintah, maka tanpa sadar para IQ 200 Sekolam ini berpendapat bahwa pemerintah RI itu PKI.	'jika membahas pki dituduh sebagai upaya untuk menyerang pemerintah, maka tanpa sadar para iq 200 sekolam ini berpendapat bahwa pemerintah ri itu pki.'
7	Jika PKI sudah tidak ada lagi di Indonesia, lantas kenapa kok banyak yang blingsatan ketika kita mau memutar film G30SPKI ???	'jika pki sudah tidak ada lagi di indonesia, lantas kenapa kok banyak yang blingsatan ketika kita mau memutar film g30spki ???'
8	Mungkin kalian sudah demikian takutnya kehilangan kekuasaan ya? Kok segala sesuatu dihubung- hubungkan dengan politik?	'mungkin kalian sudah demikian takutnya kehilangan kekuasaan ya? kok segala sesuatu dihubung- hubungkan dengan politik?'
9	Jika ada orang yang demikian gampangnya menuduh orang lain sebagai pecandu PCC dan FLAKKA,	'jika ada orang yang demikian gampangnya menuduh orang lain sebagai pecandu pcc dan flakka,

	maka mungkin dia... Ah.... Tak tega aku meneruskannya bang!	maka mungkin dia... ah.... tak tega aku meneruskannya bang!'
10	Hoax dan kebencian dari kubu mereka dipelihara, sementara dari pihak kita... ya gitu deh.	'hoax dan kebencian dari kubu mereka dipelihara, sementara dari pihak kita... ya gitu deh.'

Tabel 4.5 Data Uji yang mengalami proses *case folding*

No.	Data	Setelah proses case folding
1	MEDIA BAJINGAN & PENJAHAT SOSMED	'media bajingan & penjahat sosmed'
2	Penundaan Sidang Penista Agama dengan agenda pembacaan Rencana Tuntutan oleh JPU dengan alasan belum siap karena belum selesai diketik adalah TRAGEDI PENEGAKAN HUKUM.	'penundaan sidang penista agama dengan agenda pembacaan rencana tuntutan oleh jpu dengan alasan belum siap karena belum selesai diketik adalah tragedi penegakan hukum.'
3	Ahok untuk Kaum Zhongers alias pendukungnya saja dan mengayomi hanya kelompok pendukungnya saja.	'ahok untuk kaum zhongers alias pendukungnya saja dan mengayomi hanya kelompok pendukungnya saja.'

4.6.2 Tokenisasi

Tahap berikutnya dalam proses manualisasi ini adalah tokenisasi pada data yaitu melakukan pemecahan terhadap dokumen teks yang berupa kalimat menjadi kata yang membangun kalimat tersebut. Manualisasi tokenisasi pada data dapat dilihat pada Tabel 4.6 dan 4.7

Tabel 4.6 Data Latih yang mengalami proses tokenisasi

No.	Data hasil case folding	Setelah proses tokenisasi
1	'semuanya difitnah penebar sara dengan ujaran kebencian padahal semua anti kezaliman & berjasa besar dalam melengserkan & melongsorkan ahok si penista agama'	['semuanya', 'difitnah', 'penebar', 'sara', 'dengan', 'ujaran', 'kebencian', 'padahal', 'semua', 'anti', 'kezaliman', '&', 'berjasa', 'besar', 'dalam', 'melengserkan', '&', 'melongsorkan', 'ahok', 'si', 'agama']

		'penista', 'agama']
2	'bagi cecunguk koh napi, menebar fitnah setiap hari hukumnya adalah wajib!'	['bagi', 'cecunguk', 'koh', 'napi', ',', 'menebar', 'fitnah', 'setiap', 'hari', 'hukumnya', 'adalah', 'wajib', '!']
3	'tapi pertarungan antara pembela islam vs penista islam'	['tapi', 'pertarungan', 'antara', 'pembela', 'islam', 'vs', 'penista', 'islam']
4	'hal biasa tokoh masyarakat menemui habib rizieq, karena kalau yang mereka temui si desi atau abu janda bisa apesss dan kalah super telak lagi seperti ahok-djarot kemarin'	['hal', 'biasa', 'tokoh', 'masyarakat', 'menemui', 'habib', 'rizieq', ',', 'karena', 'kalau', 'yang', 'mereka', 'temui', 'si', 'desi', 'atau', 'abu', 'janda', 'bisa', 'apesss', 'dan', 'kalah', 'super', 'telak', 'lagi', 'seperti', 'ahok-djarot', 'kemarin']
5	'cecunguk owe makin blo'on aja. udah tau owe kalah telak selisih hampir satu juta suara masih aja mereka ngaku mayoritas dan ngajakin ngirim karangan bunga sampah begini.'	['cecunguk', 'owe', 'makin', 'blo'on', 'aja', '.', 'udah', 'tau', 'owe', 'kalah', 'telak', 'selisih', 'hampir', 'satu', 'juta', 'suara', 'masih', 'aja', 'mereka', 'ngaku', 'mayoritas', 'dan', 'ngajakin', 'ngirim', 'karangan', 'bunga', 'sampah', 'begini', '.']
6	'jika membahas pki dituduh sebagai upaya untuk menyerang pemerintah, maka tanpa sadar para iq 200 sekolam ini berpendapat bahwa pemerintah ri itu pki.'	['jika', 'membahas', 'pki', 'dituduh', 'sebagai', 'upaya', 'untuk', 'menyerang', 'pemerintah', ',', 'maka', 'tanpa', 'sadar', 'para', 'iq', '200', 'sekolam', 'ini', 'berpendapat', 'bahwa', 'pemerintah', 'ri', 'itu', 'pki', '.']
7	'jika pki sudah tidak ada lagi di indonesia, lantas kenapa kok banyak yang blingsatan ketika kita mau memutar film g30spki ???'	['jika', 'pki', 'sudah', 'tidak', 'ada', 'lagi', 'di', 'indonesia', ',', 'lantas', 'kenapa', 'kok', 'banyak', 'yang', 'blingsatan', 'ketika', 'kita', 'mau', 'memutar', 'film', 'g30spki', '?', '?', '?']
8	'mungkin kalian sudah demikian takutnya kehilangan kekuasaan ya? kok segala sesuatu dihubung-hubungkan dengan politik?'	['mungkin', 'kalian', 'sudah', 'demikian', 'takutnya', 'kehilangan', 'kekuasaan', 'ya', '?', 'kok', 'segala', 'sesuatu', 'dihubung-hubungkan',

		'dengan', 'politik', '?']
9	'jika ada orang yang demikian gampangya menuduh orang lain sebagai pecandu pcc dan flakka, maka mungkin dia... ah.... tak tega aku meneruskannya bang!'	['jika', 'ada', 'orang', 'yang', 'demikian', 'gampangya', 'menuduh', 'orang', 'lain', 'sebagai', 'pecandu', 'pcc', 'dan', 'flakka', ',', 'maka', 'mungkin', 'dia', '...', 'ah', '...', ',', 'tak', 'tega', 'aku', 'meneruskannya', 'bang', '!']
10	hoax dan kebencian dari kubu mereka dipelihara, sementara dari pihak kita... ya gitu deh.	['hoax', 'dan', 'kebencian', 'dari', 'kubu', 'mereka', 'dipelihara', ',', 'sementara', 'dari', 'pihak', 'kita', '...', 'ya', 'gitu', 'deh', '.']

Tabel 4.7 Data Uji yang mengalami proses tokenisasi

No.	Data hasil case folding	Setelah proses tokenisasi
1	'media bajingan & penjahat sosmed'	['media', 'bajingan', '&', 'penjahat', 'sosmed']
2	'penundaan sidang penista agama dengan agenda pembacaan rencana tuntutan oleh jpu dengan alasan belum siap karena belum selesai diketik adalah tragedi penegakan hukum.'	['penundaan', 'sidang', 'penista', 'agama', 'dengan', 'agenda', 'pembacaan', 'rencana', 'tuntutan', 'oleh', 'jpu', 'dengan', 'alasan', 'belum', 'siap', 'karena', 'belum', 'selesai', 'diketik', 'adalah', 'tragedi', 'penegakan', 'hukum', '.']
3	ahok untuk kaum zhongers alias pendukungnya saja dan mengayomi hanya kelompok pendukungnya saja.	['ahok', 'untuk', 'kaum', 'zhongers', 'alias', 'pendukungnya', 'saja', 'dan', 'mengayomi', 'hanya', 'kelompok', 'pendukungnya', 'saja', '.']

4.6.3 Filtering

Pada langkah *filtering*, hasil dari tokenisasi sebelumnya yang masih berisi kata-kata yang tidak berpengaruh terhadap hasil identifikasi ujaran kebencian yang biasanya disebut *stopword* akan dihilangkan. Manualisasi *filtering* pada data dapat dilihat pada Tabel 4.8 dan 4.9

Tabel 4.8 Data Latih yang mengalami proses *filtering*

No.	Data hasil tokenisasi	Setelah proses filtering
1	['semuanya', 'difitnah', 'penebar', 'sara', 'dengan', 'ujaran', 'kebencian', 'padahal', 'semua', 'anti', 'kezaliman', '&', 'berjasa', 'besar', 'dalam', 'melengserkan', '&', 'melongsorkan', 'ahok', 'si', 'penista', 'agama']	['semuanya', 'difitnah', 'penebar', 'sara', 'ujaran', 'kebencian', 'padahal', 'semua', 'anti', 'kezaliman', '&', 'berjasa', 'besar', 'melengserkan', '&', 'melongsorkan', 'ahok', 'si', 'penista', 'agama']
2	['bagi', 'cecunguk', 'koh', 'napi', ',', 'menebar', 'fitnah', 'setiap', 'hari', 'hukumnya', 'adalah', 'wajib', '!']	['cecunguk', 'koh', 'napi', ',', 'menebar', 'fitnah', 'hari', 'hukumnya', 'wajib', '!']
3	['tapi', 'pertarungan', 'antara', 'pembela', 'islam', 'vs', 'penista', 'islam']	['pembela', 'islam', 'vs', 'penista', 'islam']
4	['hal', 'biasa', 'tokoh', 'masyarakat', 'menemui', 'habib', 'rizieq', ',', 'karena', 'kalau', 'yang', 'mereka', 'temui', 'si', 'desi', 'atau', 'abu', 'janda', 'bisa', 'apesss', 'dan', 'kalah', 'super', 'telak', 'lagi', 'seperti', 'ahok-djarot', 'kemarin']	['biasa', 'tokoh', 'masyarakat', 'menemui', 'habib', 'rizieq', ',', 'kalau', 'mereka', 'temui', 'si', 'desi', 'abu', 'janda', 'apesss', 'kalah', 'super', 'telak', 'seperti', 'ahok-djarot', 'kemarin']
5	['cecunguk', 'owe', 'makin', 'blo'on', 'aja', ',', 'udah', 'tau', 'owe', 'kalah', 'telak', 'selisih', 'hampir', 'satu', 'juta', 'suara', 'masih', 'aja', 'mereka', 'ngaku', 'mayoritas', 'dan', 'ngajakin', 'ngirim', 'karangan', 'bunga', 'sampah', 'begini', '.']	['cecunguk', 'owe', 'makin', 'blo'on', 'aja', ',', 'udah', 'tau', 'owe', 'kalah', 'telak', 'selisih', 'hampir', 'satu', 'juta', 'suara', 'aja', 'ngaku', 'mayoritas', 'ngajakin', 'ngirim', 'karangan', 'bunga', 'sampah', 'begini', '.']
6	['jika', 'membahas', 'pki', 'dituduh', 'sebagai', 'upaya', 'untuk', 'menyerang', 'pemerintah', ',', 'maka', 'tanpa', 'sadar', 'para', 'iq', '200', 'sekolam', 'ini', 'berpendapat', 'bahwa', 'pemerintah', 'ri', 'itu', 'pki', '.']	['membahas', 'pki', 'dituduh', 'upaya', 'menyerang', 'pemerintah', ',', 'tanpa', 'sadar', 'iq', '200', 'sekolam', 'berpendapat', 'pemerintah', 'ri', 'pki', '.']
7	['jika', 'pki', 'sudah', 'tidak', 'ada', 'lagi', 'di', 'indonesia', ',', 'lantas', 'kenapa', 'kok', 'banyak', 'yang']	['pki', 'tidak', 'lagi', 'indonesia', ',', 'lantas', 'kok', 'banyak', 'blingsatan', 'kita', 'mau']

	'blingsatan', 'ketika', 'kita', 'mau', 'memutar', 'film', 'g30spki', '?', '?', '?']	'memutar', 'film', 'g30spki', '?', '?', '?']
8	['mungkin', 'kalian', 'sudah', 'demikian', 'takutnya', 'kehilangan', 'kekuasaan', 'ya', '?', 'kok', 'segala', 'sesuatu', 'dihubung-hubungkan', 'dengan', 'politik', '?']	['mungkin', 'kalian', 'demikian', 'takutnya', 'kehilangan', 'kekuasaan', 'ya', '?', 'kok', 'segala', 'dihubung-hubungkan', 'politik', '?']
9	['jika', 'ada', 'orang', 'yang', 'demikian', 'gampangnya', 'menuduh', 'orang', 'lain', 'sebagai', 'pecandu', 'pcc', 'dan', 'flakka', ',', ',', 'mungkin', 'dia', '...', 'ah', '...', ',', 'maka', 'mungkin', 'dia', '...', 'ah', '...', 'tak', 'tega', 'aku', 'meneruskannya', 'bang', '!']	['ada', 'orang', 'demikian', 'gampangnya', 'menuduh', 'orang', 'sebagai', 'pecandu', 'pcc', 'flakka', ',', ',', 'mungkin', 'dia', '...', 'ah', '...', ',', 'tak', 'tega', 'aku', 'meneruskannya', 'bang', '!']
10	['hoax', 'dan', 'kebencian', 'dari', 'kubu', 'mereka', 'dipelihara', ',', 'sementara', 'dari', 'pihak', 'kita', '...', 'ya', 'gitu', 'deh', '.']	['hoax', 'kebencian', 'kubu', 'dipelihara', ',', 'dari', 'pihak', 'kita', '...', 'gitu', 'deh', '.']

Tabel 4.9 Data Uji yang mengalami proses *filtering*

No.	Data hasil tokenisasi	Setelah proses filtering
1	['media', 'bajingan', '&', 'penjahat', 'sosmed']	['media', 'bajingan', '&', 'penjahat', 'sosmed']
2	['penundaan', 'sidang', 'penista', 'agama', 'dengan', 'agenda', 'pembacaan', 'rencana', 'tuntutan', 'oleh', 'jpu', 'dengan', 'alasan', 'belum', 'siap', 'karena', 'belum', 'selesai', 'diktetik', 'adalah', 'tragedi', 'penegakan', 'hukum', '.']	['penundaan', 'sidang', 'penista', 'agama', 'agenda', 'pembacaan', 'rencana', 'tuntutan', 'jpu', 'alasan', 'siap', 'belum', 'selesai', 'diktetik', 'tragedi', 'penegakan', 'hukum', '.']
3	['ahok', 'untuk', 'kaum', 'zhongers', 'alias', 'pendukungnya', 'saja', 'dan', 'mengayomi', 'hanya', 'kelompok', 'pendukungnya', 'saja', '.']	['ahok', 'kaum', 'zhongers', 'alias', 'pendukungnya', 'dan', 'mengayomi', 'kelompok', 'pendukungnya', 'saja', '.']

4.6.4 Stemming

Setelah dilakukan *filtering*, data hasil *filtering* akan diproses untuk menghilangkan imbuhan pada kata berimbuhan sehingga menghasilkan kata dasar. Untuk hasil manualisasi proses *stemming* pada data dapat dilihat pada Tabel 4.10 dan 4.11

Tabel 4.10 Data Latih yang mengalami proses *stemming*

No.	Data hasil filtering	Setelah proses stemming
1	['semuanya', 'difitnah', 'penebar', 'sara', 'dengan', 'ujaran', 'kebencian', 'padahal', 'semua', 'anti', 'kezaliman', '&', 'berjasa', 'besar', 'dalam', 'melengserkan', '&', 'melongsorkan', 'ahok', 'si', 'penista', 'agama']	['semua', 'fitnah', 'tebar', 'sara', 'ujar', 'benci', 'padahal', 'semua', 'anti', 'zalim', 'jasa', 'besar', 'lengser', 'longsor', 'ahok', 'si', 'nista', 'agama']
2	['bagi', 'cecunguk', 'koh', 'napi', ',', 'menebar', 'fitnah', 'setiap', 'hari', 'hukumnya', 'adalah', 'wajib', '!']	['cecunguk', 'koh', 'napi', 'tebar', 'fitnah', 'hari', 'hukum', 'wajib']
3	['tapi', 'pertarungan', 'antara', 'pembela', 'islam', 'vs', 'penista', 'islam']	['tarung', 'bela', 'islam', 'vs', 'nista', 'islam']
4	['hal', 'biasa', 'tokoh', 'masyarakat', 'menemui', 'habib', 'rizieq', ',', 'karena', 'kalau', 'yang', 'mereka', 'temui', 'si', 'desi', 'atau', 'abu', 'janda', 'bisa', 'apesss', 'dan', 'kalah', 'super', 'telak', 'lagi', 'seperti', 'ahok-djarot', 'kemarin']	['biasa', 'tokoh', 'masyarakat', 'temu', 'habib', 'rizieq', 'kalau', 'mereka', 'temu', 'si', 'desi', 'abu', 'janda', 'apesss', 'kalah', 'super', 'telak', 'seperti', 'ahok-djarot', 'kemarin']
5	['cecunguk', 'owe', 'makin', 'blo'on', 'aja', ':', 'udah', 'tau', 'owe', 'kalah', 'telak', 'selisih', 'hampir', 'satu', 'juta', 'suara', 'masih', 'aja', 'mereka', 'ngaku', 'mayoritas', 'dan', 'ngajakin', 'ngirim', 'karangan', 'bunga', 'sampah', 'begini', '.']	['cecunguk', 'owe', 'makin', 'blo', 'on', 'aja', 'udah', 'tau', 'owe', 'kalah', 'telak', 'selisih', 'hampir', 'satu', 'juta', 'suara', 'aja', 'ngaku', 'mayoritas', 'ngajakin', 'ngirim', 'karang', 'bunga', 'sampah', 'begini']
6	['jika', 'membahas', 'pki', 'dituduh', 'sebagai', 'upaya', 'untuk', 'menyerang', 'pemerintah', ',', 'maka', 'tanpa', 'sadar', 'para', 'iq', '200', 'sekolam', 'ini', 'berpendapat', 'bahwa', 'pemerintah', 'ri', 'itu', 'pki']	['bahas', 'pki', 'tudu', 'upaya', 'serang', 'perintah', 'tanpa', 'sadar', 'iq', '200', 'kolam', 'dapat', 'perintah', 'ri', 'pki']

	['.']	
7	['jika', 'pki', 'sudah', 'tidak', 'ada', 'lagi', 'di', 'indonesia', ',', 'lantas', 'kenapa', 'kok', 'banyak', 'yang', 'blingsatan', 'ketika', 'kita', 'mau', 'memutar', 'film', 'g30spki', '?', '?', '?']	['pki', 'tidak', 'lagi', 'indonesia', 'lantas', 'kok', 'banyak', 'blingsatan', 'kita', 'mau', 'putar', 'film', 'g30spki']
8	['mungkin', 'kalian', 'sudah', 'demikian', 'takutnya', 'kehilangan', 'kekuasaan', 'ya', '?', 'kok', 'segala', 'sesuatu', 'dihubung-hubungkan', 'dengan', 'politik', '?']	['mungkin', 'kalian', 'demikian', 'takut', 'hilang', 'kuasa', 'ya', 'kok', 'segala', 'hubung', 'politik']
9	['jika', 'ada', 'orang', 'yang', 'demikian', 'gampang', 'menuduh', 'orang', 'lain', 'sebagai', 'pecandu', 'pcc', 'dan', 'flakka', ',', 'maka', 'mungkin', 'dia', '...', 'ah', '...', '!', 'tak', 'tega', 'aku', 'meneruskannya', 'bang', '!']	['ada', 'orang', 'demikian', 'gampang', 'tuduh', 'orang', 'bagai', 'candu', 'pcc', 'flakka', 'mungkin', 'dia', 'ah', 'tak', 'tega', 'aku', 'terus', 'bang']
10	['hoax', 'dan', 'kebencian', 'dari', 'kubu', 'mereka', 'dipelihara', ',', 'sementara', 'dari', 'pihak', 'kita', '...', 'ya', 'gitu', 'deh', '.']	['hoax', 'benci', 'kubu', 'pelihara', 'dari', 'pihak', 'kita', 'gitu', 'deh']

Tabel 4.11 Data Uji yang mengalami proses *stemming*

No.	Data hasil filtering	Setelah proses stemming
1	['media', 'bajingan', '&', 'penjahat', 'sosmed']	['media', 'bajing', 'jahat', 'sosmed']
2	['penundaan', 'sidang', 'penista', 'agama', 'agenda', 'pembacaan', 'rencana', 'tuntutan', 'jpu', 'alasan', 'siap', 'belum', 'selesai', 'diktetik', 'tragedi', 'penegakan', 'hukum', '.']	['tunda', 'sidang', 'nista', 'agama', 'agenda', 'baca', 'rencana', 'tuntut', 'jpu', 'alas', 'siap', 'belum', 'selesai', 'ketik', 'tragedi', 'tega', 'hukum']
3	['ahok', 'kaum', 'zhongers', 'alias', 'pendukungnya', 'dan', 'mengayomi', 'kelompok', 'pendukungnya', 'saja', '.']	['ahok', 'kaum', 'zhongers', 'alias', 'dukung', 'dan', 'ayom', 'kelompok', 'dukung', 'saja']

4.6.5 Manualisasi *Feature Extraction*

Setelah dilakukan tahap *pre-processing* teks pada data latih dan data uji, kemudian dilakukan proses *Feature Extraction* yang mengambil nilai dari data latih dan data uji sehingga didapatkan nilai untuk tiap fitur yang kemudian dikelompokkan berdasarkan kategori fitur. Manualisasi proses *Feature Extraction* dapat dilihat pada Tabel 4.12, 4.13, dan 4.14.

Tabel 4.12 Daftar Fitur Khusus Terkait *Facebook* Hasil *Feature Extraction*

Data	F1	F2	F3	F4
D1	822	3	50	235
D2	4900	409	198	2098
D3	1100	45	26	325
D4	677	23	49	165
D5	2100	27	143	939
D6	149	0	14	510
D7	19000	82	1441	4091
D8	26000	4	1330	11123
D9	16000	1	731	1745
D10	17000	10	1556	3147
T1	3200	266	500	1181
T2	1200	112	74	505
T3	2600	908	1406	2796

Tabel 4.13 Daftar Fitur Tekstual Hasil *Feature Extraction*

Data	F5	F6	F7	F8	F9
D1	18	0	0	7	5

D2	11	1	0	1	0
D3	8	0	0	7	1
D4	27	0	0	4	0
D5	27	0	0	0	0
D6	16	0	0	4	16
D7	14	0	3	2	16
D8	10	0	1	0	12
D9	17	1	0	2	18
D10	9	0	0	0	13
T1	4	0	0	4	0
T2	17	0	0	4	13
T3	9	0	0	0	10

Tabel 4.14 Daftar Fitur Terkait *Lexicon* Hasil *Feature Extraction*

Data	F10	F11
D1	5	2
D2	2	1
D3	0	0
D4	8	7
D5	6	2
D6	5	3

D7	7	6
D8	4	4
D9	7	3
D10	6	2
T1	1	0
T2	9	5
T3	4	3

4.6.6 Manualisasi *Ensemble Feature*

Proses *Ensemble Feature* merupakan proses penggabungan fitur dari beberapa atau seluruh kategori fitur yang ada. Proses manualisasi *Ensemble Feature* ditunjukkan pada Tabel 4.15, 4.16, 4.17 dan 4.18

Tabel 4.15 Hasil *Ensemble Feature* dari Fitur Terkait *Facebook*, Tekstual, dan Fitur Terkait *Lexicon*

Data	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11
D1	822	3	50	235	18	0	0	7	5	5	2
D2	4900	409	198	2098	11	1	0	1	0	2	1
D3	1100	45	26	325	8	0	0	7	1	0	0
D4	677	23	49	165	27	0	0	4	0	8	7
D5	2100	27	143	939	27	0	0	0	0	6	2
D6	149	0	14	510	16	0	0	4	16	5	3
D7	19000	82	1441	4091	14	0	3	2	16	7	6
D8	26000	4	1330	11123	10	0	1	0	12	4	4
D9	16000	1	731	1745	17	1	0	2	18	7	3
D10	17000	10	1556	3147	9	0	0	0	13	6	2
T1	3200	266	500	1181	4	0	0	4	0	1	0
T2	1200	112	74	505	17	0	0	4	13	9	5
T3	2600	908	1406	2796	9	0	0	0	10	4	3

Tabel 4.16 Hasil *Ensemble Feature* dari Fitur Tekstual, dan Fitur Terkait *Lexicon*

Data	F5	F6	F7	F8	F9	F10	F11
D1	18	0	0	7	5	5	2
D2	11	1	0	1	0	2	1
D3	8	0	0	7	1	0	0
D4	27	0	0	4	0	8	7
D5	27	0	0	0	0	6	2
D6	16	0	0	4	16	5	3
D7	14	0	3	2	16	7	6
D8	10	0	1	0	12	4	4
D9	17	1	0	2	18	7	3
D10	9	0	0	0	13	6	2
T1	4	0	0	4	0	1	0
T2	17	0	0	4	13	9	5
T3	9	0	0	0	10	4	3

Tabel 4.17 Hasil *Ensemble Feature* dari Fitur Terkait *Facebook*, dan Fitur Terkait *Lexicon*

Data	F1	F2	F3	F4	F10	F11
D1	822	3	50	235	5	2
D2	4900	409	198	2098	2	1
D3	1100	45	26	325	0	0
D4	677	23	49	165	8	7
D5	2100	27	143	939	6	2
D6	149	0	14	510	5	3
D7	19000	82	1441	4091	7	6
D8	26000	4	1330	11123	4	4
D9	16000	1	731	1745	7	3
D10	17000	10	1556	3147	6	2

T1	3200	266	500	1181	1	0
T2	1200	112	74	505	9	5
T3	2600	908	1406	2796	4	3

Tabel 4.18 Hasil *Ensemble Feature* dari Fitur Terkait *Facebook*, dan Teksual

Data	F1	F2	F3	F4	F5	F6	F7	F8	F9
D1	822	3	50	235	18	0	0	7	5
D2	4900	409	198	2098	11	1	0	1	0
D3	1100	45	26	325	8	0	0	7	1
D4	677	23	49	165	27	0	0	4	0
D5	2100	27	143	939	27	0	0	0	0
D6	149	0	14	510	16	0	0	4	16
D7	19000	82	1441	4091	14	0	3	2	16
D8	26000	4	1330	11123	10	0	1	0	12
D9	16000	1	731	1745	17	1	0	2	18
D10	17000	10	1556	3147	9	0	0	0	13
T1	3200	266	500	1181	4	0	0	4	0
T2	1200	112	74	505	17	0	0	4	13
T3	2600	908	1406	2796	9	0	0	0	10

Setelah dihitung tiap fitur pada *Ensemble Feature*, tiap fitur-fitur tersebut akan mengalami proses normalisasi dengan jangkauan 0 sampai 1 yang kemudian dijadikan bobot untuk tiap fitur. Metode normalisasi yang dipergunakan adalah normalisasi Min-max. Perhitungan manual dari normalisasi Min-max pada *Ensemble Feature* adalah:

$$x'_i = \frac{x_i - \min_A}{\max_A - \min_A} (\text{new_max}_A - \text{new_min}_A) + \text{new_min}_A$$

$$x'_i = \frac{822 - 149}{26000 - 149} (1 - 0) + 0$$

$$x'_i = \frac{673}{25851} (1)$$

$$x'_i = 0.026033809(1)$$

$$x'_i = 0.026033809$$

Untuk hasil perhitungan manual dari normalisasi terhadap seluruh dokumen ditunjukkan pada Tabel 4.19 dan 4.20

Tabel 4.19 Hasil Normalisasi Min-Max Terhadap Fitur 1, 2, 3, 4, dan 5

No	F1	F2	F3	F4	F5
1	0.026034	0.003304	0.023346	0.006388	0.608696
2	0.183784	0.450441	0.119326	0.176401	0.304348
3	0.036788	0.049559	0.007782	0.014601	0.173913
4	0.020425	0.02533	0.022698	0	1
5	0.075471	0.029736	0.083658	0.070633	1
6	0	0	0	0.031484	0.521739
7	0.729217	0.090308	0.925422	0.358277	0.434783
8	1	0.004405	0.853437	1	0.26087
9	0.613168	0.001101	0.464981	0.144187	0.565217
10	0.651851	0.011013	1	0.27213	0.217391
11	0.118023	0.292952	0.315175	0.092718	0
12	0.040656	0.123348	0.038911	0.031028	0.565217
13	0.094813	1	0.902724	0.240099	0.217391

Tabel 4.20 Hasil Normalisasi Min-Max Terhadap Fitur 6, 7, 8, 9, 10, dan 11

F6	F7	F8	F9	F10	F11
0	0	1	0.277778	0.5	0.142857
1	0	0.142857	0	0.25	0.142857
0	0	1	0.055556	0	0
0	0	0.571429	0	1	1
0	0	0	0	0.75	0.285714
0	0	0.571429	0.888889	0.5	0.285714
0	1	0.285714	0.888889	0.625	0.571429
0	0.333333	0	0.666667	0.25	0.428571
1	0	0.285714	1	0.625	0.285714
0	0	0	0.722222	0.375	0.142857
0	0	0.571429	0	0.125	0

0	0	0.571429	0.722222	0.625	0.285714
0	0	0	0.555556	0.125	0.285714

4.6.7 Manualisasi *Support Vector Machine*

Setelah nilai dari tiap fitur hasil pembuatan *Ensemble Feature* diperoleh, nilai dari fitur-fitur tersebut kemudian akan diproses dalam proses klasifikasi *Support Vector Machine*. Fitur beserta nilai fitur dan jenis *Ensemble Feature* yang akan digunakan dalam proses *Support Vector Machine* ditunjukkan pada Tabel 4.21, 4.22, dan 4.23

Tabel 4.21 Fitur dari Data Latih 1, 2, 3, 4, dan 5

Fitur	D1	D2	D3	D4	D5
F1	0.026034	0.183784	0.036788	0.020425	0.075471
F2	0.003304	0.450441	0.049559	0.02533	0.029736
F3	0.023346	0.119326	0.007782	0.022698	0.083658
F4	0.006388	0.176401	0.014601	0	0.070633
F5	0.608696	0.304348	0.173913	1	1
F6	0	1	0	0	0
F7	0	0	0	0	0
F8	1	0.142857	1	0.571429	0
F9	0.277778	0	0.055556	0	0
F10	0.5	0.25	0	1	0.75
F11	0.142857	0.142857	0	1	0.285714

Tabel 4.22 Fitur dari Data Latih 6, 7, 8, 9, dan 10

Fitur	D6	D7	D8	D9	D10
F1	0	0.729217	1	0.613168	0.651851
F2	0	0.090308	0.004405	0.001101	0.011013
F3	0	0.925422	0.853437	0.464981	1
F4	0.031484	0.358277	1	0.144187	0.27213
F5	0.521739	0.434783	0.26087	0.565217	0.217391
F6	0	0	0	1	0
F7	0	1	0.333333	0	0
F8	0.571429	0.285714	0	0.285714	0
F9	0.888889	0.888889	0.666667	1	0.722222
F10	0.5	0.625	0.25	0.625	0.375

F11	0.285714	0.571429	0.428571	0.285714	0.142857
-----	----------	----------	----------	----------	----------

Tabel 4.23 Fitur dari Data Uji 1, 2, dan 3

Fitur	T1	T2	T3
F1	0.118023	0.040656	0.094813
F2	0.292952	0.123348	1
F3	0.315175	0.038911	0.902724
F4	0.092718	0.031028	0.240099
F5	0	0.565217	0.217391
F6	0	0	0
F7	0	0	0
F8	0.571429	0.571429	0
F9	0	0.722222	0.555556
F10	0.125	0.625	0.125
F11	0	0.285714	0.285714

Pertama-tama sebelum melakukan proses klasifikasi SVM perlu untuk mencari *hyperplane* atau fungsi $sign(h(x))$ melalui *sequential training SVM*. Dalam proses mencari *hyperplane* ini, hanya data latih saja yang akan digunakan karena telah diketahui kelas datanya. Adapun tahapan dari *sequential training SVM* adalah sebagai berikut:

1. Melakukan inisialisasi variabel yang dibutuhkan, antara lain:

$$\lambda = 0,5$$

$$\gamma = 0,005 \text{ (learning rate)}$$

$$\varepsilon = 0,0001 \text{ (epsilon)}$$

2. Melakukan inisialisasi untuk variabel $\alpha_i = 0$ dan melakukan perhitungan matriks hessian D_{ij} dengan Persamaan 2.9. Persamaan 2.9 ini menggunakan kernel *linear* yaitu: $k = (x_i, x_j) = (x_i \cdot x_j)$. Perhitungan manual kernel *linear* pada x_1 (dokumen ke 1) adalah sebagai berikut

$$x_1 = [0.03 \ 0.00 \ 0.02 \ 0.01 \ 0.61 \ 0.00 \ 0.00 \ 1.00 \ 0.28 \ 0.50 \ 0.14]$$

$$k = (x_i, x_j) = (x_i \cdot x_j)$$

$$k = (x_1, x_1) =$$

$$([0.03 \ 0.00 \ 0.02 \ 0.01 \ 0.53 \ 0.00 \ 0.00 \ 1.00 \ 0.28 \ 0.63 \ 0.29] \times$$

$$[0.03 \ 0.00 \ 0.02 \ 0.01 \ 0.61 \ 0.00 \ 0.00 \ 1.00 \ 0.28 \ 0.50 \ 0.14])$$

$$k = (x_1, x_1) = (1.72)$$

$$k = (x_1, x_1) = 1.72$$

Hasil perhitungan kernel ditunjukkan pada Tabel 4.24

Tabel 4.24 Hasil Perhitungan Kernel

	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
D1	1.72	0.48	1.12	1.82	1.03	1.43	1.23	0.58	1.29	0.58
D2	0.48	1.48	0.23	0.80	0.58	0.41	0.76	0.67	1.60	0.47
D3	1.12	0.23	1.04	0.75	0.18	0.71	0.45	0.14	0.47	0.11
D4	1.82	0.80	0.75	3.33	2.04	1.63	1.83	0.98	1.66	0.77
D5	1.03	0.58	0.18	2.04	1.66	0.98	1.23	0.79	1.21	0.69
D6	1.43	0.41	0.71	1.63	0.98	1.72	1.67	1.01	1.75	0.99
D7	1.23	0.76	0.45	1.83	1.23	1.67	4.30	3.32	2.70	2.55
D8	0.58	0.67	0.14	0.98	0.79	1.01	3.32	3.60	2.25	2.47
D9	1.29	1.60	0.47	1.66	1.21	1.75	2.70	2.25	3.49	2.02
D10	0.58	0.47	0.11	0.77	0.69	0.99	2.55	2.47	2.02	2.23

Sedangkan untuk proses dari perhitungan matrix Hessian pada dokumen ke-1 atau x_1 adalah sebagai berikut:

$$D_{ij} = y_i y_j (K(x_i, x_j) + \lambda^2)$$

$$D_{11} = y_1 y_1 (K(x_1, x_1) + \lambda^2)$$

$$D_{11} = 1.1(1.72 + 0.5^2)$$

$$D_{11} = (1.72 + 0.25)$$

$$D_{11} = 1.97$$

Untuk hasil dari perhitungan matriks Hessian pada seluruh dokumen ditunjukkan pada Tabel 4.25

Tabel 4.25 Hasil Perhitungan Matrix Hessian

	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
D1	1.97	0.73	1.37	2.07	1.28	-1.68	1.48	0.83	1.54	0.83
D2	0.73	1.73	0.48	1.05	0.83	-0.66	1.01	0.92	1.85	0.72
D3	1.37	0.48	1.29	1.00	0.43	-0.96	0.70	0.39	0.72	0.36
D4	2.07	1.05	1.00	3.58	2.29	-1.88	2.08	1.23	1.91	1.02

D5	1.28	0.83	0.43	2.29	1.91	-1.23	1.48	1.04	1.46	0.94
D6	-1.68	-0.66	-0.96	-1.88	-1.23	1.97	-1.92	-1.26	-2.00	-1.24
D7	1.48	1.01	0.70	2.08	1.48	-1.92	4.55	3.57	2.95	2.80
D8	0.83	0.92	0.39	1.23	1.04	-1.26	3.57	3.85	2.50	2.72
D9	1.54	1.85	0.72	1.91	1.46	-2.00	2.95	2.50	3.74	2.27
D10	0.83	0.72	0.36	1.02	0.94	-1.24	2.80	2.72	2.27	2.48

3. Lalu menghitung nilai E_i , $\delta\alpha_i$ dan α_i :

- a. Melakukan perhitungan nilai E_i dengan Persamaan 2.10 dari hasil perhitungan matriks Hessian. Perhitungan dari proses manual E_1 adalah sebagai berikut:

$$E_i = \sum_{j=0}^N a_i D_{ij}$$

$$E_1 = (1.97 * 0) + (0.73 * 0) + (1.37 * 0) + (2.07 * 0) + (1.28 * 0) + (-1.68 * 0) + (1.48 * 0) + (0.83 * 0) + (1.54 * 0) + (0.83 * 0)$$

$$E_1 = 0$$

Hasil perhitungan E_i pada seluruh data latih ditunjukkan pada Tabel 4.26

Tabel 4.26 Hasil Perhitungan Error

E_i	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
	0	0	0	0	0	0	0	0	0	0

- b. Setelah mendapatkan nilai E_i , dilakukan perhitungan nilai $\delta\alpha_i$ dengan Persamaan 2.11 perhitungan manual dari nilai $\delta\alpha_i$ adalah sebagai berikut:

$$\delta\alpha_i = \min\{\max[\gamma(1 - E_i); -a_i]; C - a_i\}$$

$$\delta\alpha_1 = \min\{\max[\gamma(1 - E_1); -a_1]; C - a_1\}$$

$$\delta\alpha_1 = \min\{\max[0,001(1 - 0); -0]; 1 - 0\}$$

$$\delta\alpha_1 = \min\{\max[0,001; -0]; 1\}$$

$$\delta\alpha_1 = \min\{0,001; 1\}$$

$$\delta\alpha_1 = 0,001$$

Untuk perhitungan $\delta\alpha$ pada seluruh dokumen ditunjukkan pada Tabel

Tabel 4.27 Hasil Perhitungan $\delta\alpha_i$

$\delta\alpha_i$	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
	0,001	0,001	0,001	0,001	0,001	0,001	0,001	0,001	0,001	0,001

- c. Setelah mendapatkan nilai $\delta\alpha_i$, nilai α_i dihitung dengan Persamaan 2.12. Perhitungan manual dari nilai α_i adalah sebagai berikut:

$$\alpha_i = a_i + \delta\alpha_i$$

$$\alpha_1 = a_1 + \delta\alpha_1$$

$$\alpha_1 = 0 + 0,001$$

$$\alpha_1 = 0,001$$

Hasil perhitungan α_i baru ditunjukkan pada Tabel 4.28

Tabel 4.28 Hasil Perhitungan α Baru

	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
α lama	0	0	0	0	0	0	0	0	0	0
α baru	0,001	0,001	0,001	0,001	0,001	0,001	0,001	0,001	0,001	0,001

4. Setelah itu ulangi langkah ketiga hingga mencapai konvergensi yaitu $\text{Max}(|\delta\alpha|) < \epsilon$. Maka hasil perhitungan manual dari E_i , $\delta\alpha_i$ dan α_i pada saat mencapai konvergensi ditunjukkan pada Tabel 4.29, 4.30, dan 4.31

Tabel 4.29 Hasil Perhitungan Error

E_i	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
	1	1	0.76	1	1	-10.86	1	1	1	1

Tabel 4.30 Hasil Perhitungan $\delta\alpha_i$

$\delta\alpha_i$	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
	4E-05	6E-05	0	1E-05	4E-05	0	5E-06	1E-05	7E-06	2E-05

Tabel 4.31 Hasil Perhitungan α

	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
α	0.09	0.10	0.13	0.07	0.09	1	0.05	0.06	0.06	0.07

5. Dari proses sebelumnya diperoleh nilai α akhir dan *support vector* dari data yang diproses, yaitu data yang memiliki nilai $\alpha > 0$. Hasil *support vector* ditunjukkan pada Tabel 4.32

Tabel 4.32 Hasil Perhitungan *Support Vector*

	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
α	0.09	0.10	0.13	0.07	0.09	1	0.05	0.06	0.06	0.07

6. Setelah itu, dilakukan perhitungan terhadap nilai *bias* b yang diperoleh dengan Persamaan 2.13. Untuk penentuan x^+ dan x^- dipilih dokumen teks yang memiliki nilai α terbesar dari masing-masing kelas sentiment. Pada perhitungan ini yang terpilih untuk x^+ adalah D3 dan x^- adalah D6. Proses perhitungan ditunjukkan pada Tabel 4.33

Tabel 4.33 Proses Perhitungan *bias*

x_i	$K(x_i, x^+)$	$K(x_i, x^-)$	$\alpha_i \gamma_i K(x_i, x^+)$	$\alpha_i \gamma_i K(x_i, x^-)$
D1	1.12	1.43	0.10	0.14
D2	0.23	0.41	0.03	0.05
D3	1.04	0.71	0.18	0.12
D4	0.75	1.63	0.05	0.11
D5	0.18	0.98	0.02	0.09
D6	0.71	1.72	-0.71	-1.72
D7	0.45	1.67	0.02	0.09
D8	0.14	1.01	0.01	0.06
D9	0.47	1.75	0.03	0.10
D10	0.11	0.99	0.01	0.08
			$\sum_{i=0}^N \alpha_i \gamma_i K(x_i, x^+)$	$\sum_{i=0}^N \alpha_i \gamma_i K(x_i, x^-)$
			-0.26	-0.88

Setelah didapatkan nilai $\sum_{i=0}^N \alpha_i \gamma_i K(x_i, x^+)$ dan nilai $\sum_{i=0}^N \alpha_i \gamma_i K(x_i, x^-)$ nilai tersebut kemudian disubstitusikan ke dalam Persamaan *bias*:

$$b = -\frac{1}{2} \left(\sum_{i=0}^N \alpha_i \gamma_i K(x_i, x^+) + \sum_{i=0}^N \alpha_i \gamma_i K(x_i, x^-) \right)$$

$$b = -\frac{1}{2} (-0.26 - 0.88)$$

$$b = 0.57$$

7. Terakhir, kita akan melakukan pengujian terhadap SVM dengan memasukkan data uji ke dalam *hyperplane* untuk mengetahui masuk ke kelas manakah data uji tersebut. Untuk melakukan klasifikasi data uji, digunakan Persamaan 2.14. Apabila hasil dari $\text{sign}(h(x))$ adalah kurang dari 0 artinya dokumen teks tersebut tidak mengandung ujaran kebencian. Namun, apabila hasilnya lebih dari 0, dokumen teks tersebut mengandung ujaran kebencian. Berikut adalah perhitungan manual dari proses pengujian SVM untuk data uji. Hasil perhitungan manual identifikasi SVM dapat dilihat pada Tabel 4.34

Tabel 4.34 Hasil Perhitungan Identifikasi SVM

$\alpha_i \gamma_i K(x, x_i)$	T1	T2	T3
D1	0.06	0.13	0.04
D2	0.03	0.05	0.08
D3	0.08	0.10	0.02
D4	0.03	0.12	0.05
D5	0.01	0.10	0.05
D6	-0.39	-1.66	-0.76
D7	0.04	0.09	0.10
D8	0.03	0.06	0.11
D9	0.03	0.10	0.08
D10	0.03	0.07	0.12
$\sum_{i=0}^N \alpha_i \gamma_i K(x, x_i)$	-0.05	-0.83	-0.13

$\text{sign}(h(x)) = \sum_{i=0}^N \alpha_i \gamma_i K(x, x_i) + b$	0.58	-0.20	0.49
Kelas Data	Ujaran Kebencian	Bukan Ujaran Kebencian	Bukan Ujaran Kebencian

4.7 Perancangan Pengujian

Pada sistem ini terdapat dua jenis pengujian yaitu pengujian pengaruh parameter *Support Vector Machine* dan pengujian jenis *Ensemble Feature*

4.7.1 Perancangan Pengujian Pengaruh Parameter *Support Vector Machine*

Pada pengujian pengaruh parameter *Support Vector Machine* terdapat beberapa jenis pengujian, yaitu pengujian nilai *lambda*, nilai *learning rate*(γ), dan nilai *epsilon*(ϵ). Pengujian ini dilakukan untuk mendapatkan nilai yang optimal dari tiap parameter *Support Vector Machine* untuk mendapatkan hasil yang terbaik. Perancangan pengujian nilai *lambda*, *learning rate*(γ), dan nilai *epsilon*(ϵ) ditunjukkan pada Tabel 4.35, Tabel 4.36, dan Tabel 4.37.

Tabel 4.35 Perancangan Pengujian Nilai *lambda*(λ)

	Nilai Lambda(λ)					Rata-rata
	0.5	0.7	0.9	1.1	1.3	
<i>Accuracy</i>						
<i>Precision</i>						
<i>Recall</i>						
<i>F-Measure</i>						

Tabel 4.36 Perancangan Pengujian Nilai *learning rate*(γ)

	Nilai Learning Rate(γ)					Rata-Rata
	0.0001	0.0005	0.001	0.005	0.01	
<i>Accuracy</i>						
<i>Precision</i>						
<i>Recall</i>						

<i>F-Measure</i>						
------------------	--	--	--	--	--	--

Tabel 4.37 Perancangan Pengujian Nilai *epsilon*(ϵ)

	Nilai <i>Epsilon</i> (ϵ)					Rata-rata
	0.00005	0.0001	0.0005	0.001	0.005	
<i>Accuracy</i>						
<i>Precision</i>						
<i>Recall</i>						
<i>F-Measure</i>						

4.7.2 Perancangan Pengujian Pengaruh Jenis *Ensemble Feature*

Pengujian jenis *Ensemble Feature* terdiri dari pengujian jenis *Ensemble Feature* dan akurasi serta precision dan recall hasil identifikasi dari sistem. Pengujian ini dilakukan untuk menentukan jenis *Ensemble Feature* mana yang lebih cocok untuk mendapatkan akurasi maksimal. Perancangan untuk pengujian jenis *Ensemble Feature* ditunjukkan pada Tabel 4.38

Tabel 4.38 Perancangan Pengujian Jenis *Ensemble Feature*

	Jenis <i>Ensemble</i>							Rata-rata
	<i>Facebook, Tekstual, Lexicon</i>	<i>Facebook, Tekstual</i>	<i>Facebook, Lexicon</i>	<i>Tekstual, Lexicon</i>	<i>Facebook</i>	<i>Tekstual</i>	<i>Lexicon</i>	
<i>Accuracy</i>								
<i>Precision</i>								
<i>Recall</i>								
<i>F-Measure</i>								

BAB 5 IMPLEMENTASI

Bab ini akan menjelaskan mengenai batasan-batasan yang terdapat dalam implementasi dan proses implementasi sistem berdasarkan perancangan pada bab sebelumnya.

5.1 Batasan Implementasi

Batasan implementasi merupakan sejumlah batasan yang dipenuhi oleh sistem berdasarkan bab-bab sebelumnya. Dengan tujuan agar sistem dapat fokus dalam proses menyelesaikan masalah yang terjadi dan tidak keluar dari tujuan utama penelitian. Beberapa batasan dalam proses implementasi pada penelitian ini adalah sebagai berikut:

1. Identifikasi ujaran kebencian pada *Facebook* diimplementasikan dengan bahasa pemrograman Python.
2. Metode yang dipergunakan untuk proses identifikasi adalah *Ensemble Feature* dan *Support Vector Machine*.
3. Data yang dipergunakan berupa kalimat dalam post yang diambil dari beberapa halaman public pada *Facebook* yang kemudian disimpan dalam fail txt.
4. Selain menggunakan data berupa kalimat dalam post, digunakan juga parameter khusus yang dimiliki *Facebook* antara lain: jumlah *like*, jumlah *angry reaction*, jumlah *comment*, dan jumlah *share* yang kemudian disimpan dalam fail xls.
5. Hasil keluaran identifikasi pada penelitian ini yaitu ujaran kebencian dan buka ujaran kebencian.

5.2 Implementasi Sistem

Pada bagian ini akan diuraikan tahapan identifikasi ujaran kebencian pada *Facebook* dengan metode *Ensemble Feature* dan *Support Vector Machine*. Tahapan implementasi sistem diuraikan berdasar tahap perancangan sistem pada bab sebelumnya dan meliputi potongan Kode Program untuk setiap proses identifikasi.

Tahapan proses identifikasi untuk penelitian ini terdiri dari *pre-processing* teks (meliputi *case folding*, tokenisasi, *filtering*, dan *stemming*), perhitungan fitur (meliputi perhitungan fitur khusus terkait *Facebook*, fitur tekstual, dan fitur terkait *lexicon*) dan identifikasi *Support Vector Machine* yang digunakan untuk mengidentifikasi apakah kalimat dalam sebuah post termasuk ujaran kebencian atau bukan ujaran kebencian. Fungsi-fungsi yang digunakan dalam sistem ditunjukkan pada Tabel 5.1.

Tabel 5.1 Daftar fungsi sistem identifikasi ujaran kebencian

No.	Proses	Nama Fungsi	Keterangan
1	<i>Pre-processing</i>	<i>translate()</i>	Fungsi untuk menghilangkan tanda baca pada teks
		<i>stem()</i>	Fungsi untuk melakukan <i>stemming</i> pada seluruh kalimat dalam dokumen termasuk proses <i>case folding</i> , <i>filtering</i> , dan <i>tokenisasi</i>
2	Ekstraksi Fitur	<i>hitung()</i>	Fungsi untuk melakukan ekstraksi terhadap fitur-fitur yang digunakan dalam proses identifikasi
		<i>list_negatif()</i>	Fungsi untuk membaca fail txt yang berisikan kata negatif dan membuat list dari fail tersebut
		<i>list_positif()</i>	Fungsi untuk membaca fail txt yang berisikan kata positif dan membuat list dari fail tersebut
		<i>import_data()</i>	Fungsi untuk membaca fail excel yang berisi fitur terkait <i>Facebook</i> dari data yang akan digunakan
		<i>import_kelas()</i>	Fungsi untuk membaca fail excel yang berisi kelas dari data yang akan digunakan
		<i>import_data()</i>	Fungsi untuk membaca fail excel yang berisi fitur terkait <i>Facebook</i> dari data yang akan digunakan
		<i>hitung_min_max()</i>	Fungsi untuk melakukan normalisasi min max terhadap data yang akan digunakan untuk proses identifikasi
3	<i>Ensemble Feature</i>	<i>ensemble()</i>	Fungsi untuk melakukan penggabungan fitur menjadi satu dengan menentukan jenis <i>Ensemble Feature</i> mana yang akan digunakan serta membentuk matriks dan matriks transpos dari fitur

4	<i>Support Vector Machine</i>	<i>kernel()</i>	Fungsi untuk melakukan perhitungan kernel <i>linear</i> dari matriks <i>Ensemble Feature</i>
		<i>hessianmatrix()</i>	Fungsi untuk melakukan perhitungan <i>matrix hessian</i> pada matrix data latih, kernel, dan kelasnya
		<i>SVMSequential()</i>	Fungsi yang dipergunakan untuk melakukan training pada SVM dengan cara mencari nilai error (E), $\delta\alpha$, dan α sampai mencapai konvergensi untuk mendapatkan nilai <i>Support Vector</i> yang dijadikan acuan untuk penentuan <i>Hyperplane</i> dari <i>Support Vector Machine</i>
		<i>supportvector()</i>	Fungsi yang dipergunakan untuk menentukan nilai dari <i>support vector</i> berdasarkan nilai α terakhir yang diperoleh
		<i>bias()</i>	Fungsi untuk perhitungan nilai <i>bias</i> pada data latih berdasarkan <i>support vector</i> , matriks data latih, dan kelas
		<i>identifikasi()</i>	Fungsi untuk melakukan identifikasi pada data uji berdasarkan matriks data latih, kelas data latih dan <i>support vector</i>

5.3 Pre-processing teks

Pre-processing teks adalah tahapan pertama dari identifikasi yang berfungsi untuk mengolah teks dari data menjadi kata-kata atau fitur yang kemudian diolah dalam proses identifikasi nantinya. Tahapan implementasi dalam *pre-processing* teks meliputi 2 proses yaitu stopword, stemming proses, dan tokenisasi.

5.3.1 Implementasi Stopword

Stopword merupakan proses pertama dalam tahapan *pre-processing* teks. Pada tahap ini dilakukan 2 proses yaitu proses *case folding* dan *filtering*. Proses ini akan mengubah seluruh huruf dalam kalimat menjadi huruf kecil dan

menghilangkan *stopword* serta tanda baca dalam kalimat. Proses *stopword* ditunjukkan pada Kode Program 5.1.

Algoritme 1 : <i>stopword</i>	
1	<code>def stopwords(a):</code>
2	<code>factory = StopWordRemoverFactory()</code>
3	<code>stopword = factory.create_stop_word_remover()</code>
4	<code>tandabaca = a.translate(None, string.punctuation)</code>
5	<code>stop = stopwords.remove(tandabaca)</code>
6	<code>return stop</code>

Kode Program 5.1 Implementasi *stopword*

Penjelasan dari potongan Kode Program 5.1 mengenai proses implementasi *stopword* diuraikan sebagai berikut:

1. Baris ke 2 dan 3 menunjukkan proses dari pembuatan *stopword remover* dari *library* Sastrawi
2. Baris ke 4 menunjukkan proses menghapus tanda baca pada kalimat dengan menggunakan fungsi *translate*
3. Baris ke 5 menunjukkan proses penghapusan *stopword* pada kalimat dengan *stopword remover* yang telah dibuat sebelumnya pada hasil penghapusan tanda baca.

5.3.2 Implementasi Stemming Proses

Stemming proses dilakukan *Stemming* merupakan salah satu proses dari *pre-processing* teks yang mengubah kata berimbuhan menjadi kata dasar. Proses ini dilakukan setelah proses *stopword* selesai dilakukan. Proses *stemming* proses ditunjukkan pada Kode Program 5.2.

Algoritme 2 : <i>stemming</i> proses	
1	<code>def stemming_proses(a):</code>
2	<code>factory = StemmerFactory()</code>
3	<code>stemmer = factory.create_stemmer()</code>
4	<code>stemming = stemmer.stem(a)</code>
5	<code>return stemming</code>

Kode Program 5.2 Implementasi stemming proses

Penjelasan potongan Kode Program 5.2 mengenai proses implementasi *stemming* proses diuraikan sebagai berikut:

1. Baris ke 2 dan 3 menunjukkan proses dari pembuatan *stemmer* dari *library* Sastrawi

- Baris ke 4 menunjukkan proses *stemming* pada kalimat dengan menggunakan *stemmer* yang telah dibuat sebelumnya.

5.3.3 Implementasi Tokenisasi

Tokenisasi dilakukan setelah proses *stemming* selesai dilakukan. Tahapan tokenisasi ini bertujuan untuk memecah kalimat menjadi kata-kata. Proses tokenisasi ditunjukkan pada Kode Program 5.3.

Algoritme 3 : tokenisasi	
1	def tokenisasi(a):
2	stem_token = a.split()
3	return stem_token

Kode Program 5.3 Implementasi tokenisasi

Penjelasan potongan Kode Program 5.3 mengenai proses implementasi tokenisasi diuraikan sebagai berikut:

- Baris ke 2 menunjukkan proses pemecahan kalimat dari dokumen dengan pemisah berupa spasi.

5.4 Ekstraksi Fitur

Ekstraksi fitur merupakan proses untuk memperoleh nilai dari fitur yang dibutuhkan untuk proses selanjutnya dari data. Tahapan implementasi ekstraksi fitur hanya terdiri dari empat proses yaitu proses Hitung yang akan melakukan ekstraksi pada fitur tekstual dan fitur *lexicon* yang dibutuhkan pada proses, proses *import* kelas yang akan melakukan import data kelas dari fail xls, proses *import* data yang akan melakukan *import* data fitur khusus terkait *Facebook* dari fail xls, dan proses hitung *min max* untuk melakukan normalisasi *min max* pada data.

5.4.1 Implementasi Hitung

Hitung atau proses perhitungan fitur dalam data merupakan proses yang akan melakukan ekstraksi terhadap fitur tekstual dan fitur *lexicon* dari data. Proses hitung ditunjukkan pada Kode Program 5.4.

Algoritme 4 : hitung	
1	def hitung(a):
2	f5 = 0
3	f6 = 0
4	f7 = 0
5	f8 = 0
6	f9 = 0
7	f10 = 0

8	f11 = 0
9	ensemble_teks = []
10	ensemble_lex = []
11	kata_negatif = list_negatif()
12	kata_positif = list_positif()
13	split = a.split()
14	for word in split:
15	f6 = word.count("!")
16	f7 = word.count("?")
17	
18	f8 = sum(word.isupper() for word in split)
19	f9 = sum(word.islower() for word in split)
20	stop = stopword(a)
21	stemming = stemming_proses(stop)
22	stem_token = tokenisasi(stemming)
23	for word in stem_token:
24	if word in kata_negatif:
25	f10+=1
26	if word in kata_positif:
27	f11+=1
28	
29	
30	
31	f5 = len(stemming.split())
32	ensemble_teks.append(float(f5))
33	ensemble_teks.append(float(f6))
34	ensemble_teks.append(float(f7))
35	ensemble_teks.append(float(f8))
36	ensemble_teks.append(float(f9))
37	
38	ensemble_lex.append(float(f10))
39	ensemble_lex.append(float(f11))
40	return ensemble_teks,ensemble_lex

Kode Program 5.4 Implementasi hitung

Penjelasan potongan Kode Program 5.4 mengenai proses implementasi hitung diuraikan sebagai berikut:

1. Baris ke 12 menunjukkan pemanggilan metode list negatif untuk menghitung kata negatif yang sebelumnya telah diambil dari fail negatif.txt untuk diubah menjadi *array*. Cuplikan dari isi fail negatif.txt ditunjukkan pada Gambar 5.1. Seluruh isi dari kata negatif terlihat pada kolom pertama

```

1 abnormal
2 absurd
3 acak
4 acak-acakan
5 acuh
6 acuh tak acuh
7 adiktif
8 adil
9 agresif
10 agresif
11 agresor

```

Gambar 5.1 Cuplikan isi *file* negatif.txt

2. Baris ke 13 menunjukkan pemanggilan metode list positif untuk menghitung kata positif yang sebelumnya telah diambil dari fail positif.txt untuk diubah menjadi *array*. Cuplikan dari isi fail positif.txt ditunjukkan pada Gambar 5.2. Seluruh isi dari kata positif terlihat pada kolom pertama

```

1 a+
2 acungan jempol
3 adaptif
4 adil
5 afinitas
6 afirmasi
7 agilely
8 agung
9 ahli
10 ahlinya
11 ajaib

```

Gambar 5.2 Cuplikan isi *file* positif.txt

3. Baris ke 16 dan 17 melakukan perhitungan terhadap jumlah tanda seru dan tanda tanya dalam kalimat
4. Baris ke 19 dan 20 menunjukkan proses perhitungan huruf kecil dan huruf kapital dalam kalimat
5. Baris ke 21 menunjukkan proses pemanggilan metode *pre-processing* teks untuk melakukan *pre-processing* pada kalimat
6. Baris ke 22 menunjukkan proses pemanggilan metode tokenisasi untuk melakukan tokenisasi pada kalimat
7. Baris ke 24 sampai 27 menunjukkan proses perhitungan kata positif dan negatif dalam kalimat
8. Baris ke 31 menunjukkan proses perhitungan jumlah kata dalam kalimat

5.4.2 Implementasi Import Data

Import data merupakan proses untuk mengambil data fitur terkait *Facebook* untuk data latih dari fail xls. Proses import data sendiri diperlukan untuk membuat *Ensemble Feature* pada tahap berikutnya. Proses import data dapat dilihat pada Kode Program 5.5.

Algoritme 5 : import_data	
1	def import_data():
2	
3	book = open_workbook("data latih.xlsx")
4	sheet = book.sheet_by_index(0)
5	
6	
7	data = []
8	
9	for row in range(1, 151):
10	row1 = []
11	for col in range(1,5):
12	row1.append(sheet.cell_value(row, col))
13	data.append(row1)
14	return data

Kode Program 5.5 Implementasi import data

Penjelasan potongan Kode Program 5.5 mengenai proses implementasi import data diuraikan sebagai berikut:

1. Baris ke 3 menunjukkan proses pembukaan fail xls workbook dengan nama fail statistik data latih.xls
2. Baris ke 4 menunjukkan sheet yang digunakan dalam proses import data
3. Baris ke 9 sampai baris ke 13 menunjukkan proses pengambilan data dari fail xls

5.4.3 Implementasi Import Kelas

Import kelas merupakan proses untuk mengambil data kelas untuk data latih dari fail xls. Data kelas yang telah diambil kemudian akan digunakan untuk proses identifikasi dengan *Support Vector Machine*. Proses import kelas dapat dilihat pada Kode Program 5.6.

Algoritme 6 : import_kelas	
1	def import_kelas():
2	

3	<code>book = open_workbook("kelas data latih.xlsx")</code>
4	<code>sheet = book.sheet_by_index(0)</code>
5	
6	
7	<code>data = []</code>
8	
9	<code>for row in range(1, 151):</code>
10	<code>row1 = []</code>
11	<code>for col in range(1,5):</code>
12	<code>data.append(sheet.cell_value(row, 1))</code>
13	<code>return data</code>

Kode Program 5.6 Implementasi import kelas

Penjelasan potongan Kode Program 5.6 mengenai proses implementasi import data diuraikan sebagai berikut:

1. Baris ke 3 menunjukkan proses pembukaan fail xls workbook dengan nama fail kelas data asli.xls
2. Baris ke 4 menunjukkan sheet yang digunakan dalam proses import data
3. Baris ke 9 sampai baris ke 12 menunjukkan proses pengambilan data dari fail xls

5.4.4 Implementasi Hitung Min Max

Hitung min max adalah proses untuk melakukan normalisasi min max terhadap data fitur untuk data latih dan data uji. Normalisasi min max dilakukan karena range data yang terlalu besar (terutama untuk fitur terkait *Facebook*) dan untuk mempermudah proses identifikasi oleh *Support Vector Machine*. Proses hitung min max dapat dilihat pada Kode Program 5.7.

Algoritme 7 : hitung_min_max	
1	<code>def hitung_min_max(teks):</code>
2	<code>x = np.array(teks)</code>
3	<code>maxi = np.max(x,axis=0)</code>
4	<code>mini = np.min(x,axis=0)</code>
5	<code>normalisasi = []</code>
6	<code>for x in range(0, len(teks)):</code>
7	<code>norm = []</code>
8	<code>normal = 0</code>
9	<code>for y in range(0,len(teks[0])):</code>
10	<code>if (maxi[y]-mini[y])>0:</code>

11	<code>normal = ((teks[x][y]-mini[y])/(maxi[y]-mini[y]))*((1-0)+0)</code>
12	<code>else :</code>
13	<code>normal = 0</code>
14	<code>norm.append(normal)</code>
15	<code>normalisasi.append(norm)</code>
16	<code>return normalisasi</code>

Kode Program 5.7 Implementasi hitung min max

Penjelasan potongan Kode Program 5.7 mengenai proses implementasi hitung min max diuraikan sebagai berikut:

1. Baris ke 3 menunjukkan nilai maksimal dari tiap fitur
2. Baris ke 4 menunjukkan nilai minimal dari tiap fitur
3. Baris ke 9 sampai baris ke 13 menunjukkan proses normalisasi min max untuk tiap data fitur dari tiap kalimat

5.5 Ensemble Feature

Ensemble Feature adalah proses menggabungkan beberapa kategori fitur menjadi sebuah kumpulan fitur sesuai kebutuhan. *Ensemble Feature* kemudian akan digunakan untuk proses identifikasi *Support Vector Machine* yang akan melakukan identifikasi terhadap data uji. Dalam proses *Ensemble Feature* terdapat 1 proses saja yaitu proses *Ensemble*.

5.5.1 Implementasi Ensemble

Ensemble merupakan sebuah proses untuk menggabungkan beberapa kategori fitur menjadi satu matriks fitur dan membuat matriks tranposisinya. Pembentukan matriks dan tranposisi dari matriks fitur diperlukan untuk proses *Support Vector Machine* yang seringkali menggunakan perkalian matriks. Untuk proses ensemble dapat dilihat pada Kode Program 5.8.

Algoritme 8 : ensemble	
1	<code>def ensemble(fb, teks, lex, tipe):</code>
2	<code>fitur = []</code>
3	<code>if (tipe == 1):</code>
4	<code>for i in range(0, len(fb)):</code>
5	<code>fitur = [a+b+c for a,b,c in zip(fb, teks, lex)]</code>
6	<code>if (tipe == 2):</code>
7	<code>for i in range(0, len(fb)):</code>
8	<code>fitur = [a+b for a,b in zip(fb, teks)]</code>
9	<code>if (tipe == 3):</code>
10	<code>for i in range(0, len(teks)):</code>

11	fitur = [a+b for a,b in zip(teks,lex)]
12	if (tipe == 4):
13	for i in range(0,len(fb)):
14	fitur = [a+b for a,b in zip(fb,lex)]
15	if (tipe == 5):
16	for i in range(0,len(fb)):
17	fitur = fb
18	if (tipe == 6):
19	for i in range(0,len(fb)):
20	fitur = teks
21	if (tipe == 7):
22	for i in range(0,len(fb)):
23	fitur = lex
24	
25	transpos = []
26	for i in range(0,len(fitur[0])):
27	row = []
28	for j in range(0,len(fitur)):
29	row.append(fitur[j][i])
30	transpos.append(row)
31	return fitur,transpos

Kode Program 5.8 Implementasi *ensemble*

Penjelasan potongan Kode Program 5.8 mengenai proses implementasi *ensemble* diuraikan sebagai berikut:

1. Baris ke 3 sampai ke 23 menunjukkan proses pembuatan *ensemble feature* berdasarkan jenis *ensemble feature* yang dipilih.
2. Baris ke 26 sampai ke 30 menunjukkan proses pembuatan matriks transpos dari fitur.

5.6 Support Vector Machine

Metode *Support Vector Machine* merupakan metode inti pada proses identifikasi ujaran kebencian. *Support Vector Machine* akan mencari *hyperplane* untuk memisahkan data latih yang berupa ujaran kebencian atau bukan ujaran kebencian. Proses identifikasi dengan metode *Support Vector Machine* yang meliputi proses perhitungan kernel, matriks *hessian*, *sequential training SVM*, penentuan *support vector*, *bias*, dan identifikasi atau penentuan jenis kelas terhadap data yang diujikan.

5.6.1 Implementasi Perhitungan Kernel

Perhitungan kernel adalah tahapan pertama yang dilakukan dalam proses *Support Vector Machine*. Proses ini dilakukan untuk mendapatkan nilai matriks kernel yang akan dipergunakan dalam proses-proses selanjutnya. Kernel yang dipergunakan dalam proses penelitian ini adalah kernel *Linear*. Implementasi perhitungan kernel dapat dilihat pada Kode Program 5.9.

Algoritme 9 : kernel	
1	def kernel(kelas,data,transpos):
2	kerneldata = []
3	for i in range(0,len(kelas)):
4	row = []
5	for j in range(0,len(kelas)):
6	jumlah = 0
7	for k in range(0,len(data[0])):
8	kali = data[i][k]*transpos[k][j]
9	jumlah = jumlah+(kali)
10	kernel = jumlah
11	row.append(kernel)
12	kerneldata.append(row)
13	return kerneldata

Kode Program 5.9 Implementasi perhitungan Kernel

Penjelasan potongan Kode Program 5.9 mengenai proses implementasi perhitungan kernel diuraikan sebagai berikut:

1. Baris ke 7 sampai ke 9 menunjukkan proses perkalian matriks untuk tiap fitur pada data.
2. Baris ke 10 menunjukkan perhitungan kernel untuk tiap data dengan menggunakan kernel *linear*.

5.6.2 Implementasi Perhitungan Matriks *Hessian*

Perhitungan matriks *Hessian* merupakan tahapan setelah tahap perhitungan kernel dalam proses *Support Vector Machine*. Proses ini dilakukan untuk memperoleh nilai dari matriks *Hessian* berdasar dari kelas data latih, hasil perhitungan kernel dan lambda. Implementasi perhitungan matriks *Hessian* ditunjukkan pada Kode Program 5.10.

Algoritme 10 : hessianmatrix	
1	def hessianmatrix(kernel,kelas):
2	lam = 0.5

3	<code>hessian_matrix = []</code>
4	<code>for i in range(0,len(kernel)):</code>
5	<code>row = []</code>
6	<code>for j in range(0,len(kernel[i])):</code>
7	<code>hessian = (kelas[i]*kelas[j])*(kernel[i][j]+(lam*lam))</code>
8	<code>row.append(hessian)</code>
9	<code>hessian_matrix.append(row)</code>
10	<code>return hessian_matrix</code>

Kode Program 5.10 Implementasi perhitungan matriks *Hessian*

Penjelasan potongan Kode Program 5.10 mengenai proses implementasi perhitungan matriks *hessian* diuraikan sebagai berikut:

1. Baris ke 2 menunjukkan nilai dari variable lambda yang akan digunakan.
2. Baris ke 7 sampai ke 8 menunjukkan proses perhitungan dari matriks *hessian* berdasarkan kernel, kelas data, dan lambda.

5.6.3 Implementasi Perhitungan *Sequential Training SVM* dan Penentuan *Support Vector*

Tahap selanjutnya dari proses *Support Vector Machine* adalah proses perhitungan *sequential training SVM* dan Penentuan *support vector*. Proses ini dilakukan untuk menentukan *support vector* yang akan digunakan untuk pembuatan *hyperplane* yang berfungsi untuk memisahkan antara data ujaran kebencian dan bukan ujaran kebencian. Implementasi perhitungan *sequential training SVM* dan penentuan *support vector* dapat dilihat pada Kode Program 5.11.

Algoritme 11 : SVMSequential & support_vector	
1	<code>def SVMSequential(hessian,learn_rate):</code>
2	<code>slack = 1</code>
3	<code>epsilon = 0.0001</code>
4	<code>E = []</code>
5	<code>thoalpha = []</code>
6	<code>alpha = []</code>
7	<code>rowa = []</code>
8	<code>for x in range(0,len(hessian)):</code>
9	<code>rowa.append(0)</code>
10	<code>alpha.append(rowa)</code>
11	<code>for i in range(0,500):</code>
12	<code>epsi = []</code>

13	thoa = []
14	a = []
15	for j in range(0,len(hessian)):
16	jumlah = 0
17	for k in range(0,len(hessian[0])):
18	jumlah = jumlah + (hessian[j][k]* alpha[-1][j])
19	epsi.append(jumlah)
20	E.append(epsi)
21	for j in range(0,len(E[-1])):
22	total = min(max(learn_rate*(1-E[-1][j]),-alpha[-1][j]),slack-alpha[-1][j])
23	thoa.append(total)
24	if max(thoa)<epsilon:
25	break
26	thoalpha.append(thoa)
27	for j in range(0,len(thoalpha[-1])):
28	hasil = alpha[-1][j]+thoalpha[-1][j]
29	a.append(hasil)
30	alpha.append(a)
31	return E[-1],thoalpha[-1],alpha[-1]
32	
33	def support_vector (alpha):
34	alpha_new = []
35	for i in range(0,len(alpha)):
36	if (alpha[i]>0):
37	alpha_new.append(alpha[i])
38	return alpha_new

Kode Program 5.11 Implementasi *sequential training SVM* dan penentuan *support vector*

Penjelasan potongan Kode Program 5.11 mengenai proses implementasi *sequential training SVM* dan penentuan *support vector* diuraikan sebagai berikut:

1. Baris ke 2 dan ke 3 menunjukkan nilai dari variabel *slack* dan *epsilon* yang akan digunakan dalam proses perhitungan.
2. Baris ke 8 sampai ke 10 menunjukkan proses inisialisasi α awal untuk seluruh data latih sama dengan 0.
3. Baris ke 11 menunjukkan batas iterasi yang akan dilakukan oleh *sequential training SVM*.

4. Baris ke 17 sampai ke 19 menunjukkan proses perhitungan *Error* yang kemudian disimpan pada *array* di baris ke 20.
5. Baris ke 21 sampai ke 22 menunjukkan proses perhitungan $\delta\alpha$ yang kemudian disimpan pada *array* di baris ke 23.
6. Baris ke 27 sampai ke 29 menunjukkan proses perhitungan α baru yang kemudian disimpan pada baris ke 30.
7. Baris ke 35 sampai ke 37 menunjukkan proses penentuan *support vector* berdasarkan nilai α terakhir.

5.6.4 Implementasi Perhitungan *bias*

Perhitungan *bias* adalah tahap berikutnya dari proses *Support Vector Machine*. Perhitungan *bias* dilakukan setelah *support vector* ditentukan. *Bias* dihitung dengan menggunakan nilai *support vector* terbesar dari masing-masing kelas data dan digunakan untuk membentuk *hyperplane*. Implementasi perhitungan *bias* ditunjukkan pada Kode Program 5.12.

Algoritme 12 : <i>bias</i>	
1	def <i>bias</i> (<i>data</i> , <i>support_vector</i> , <i>kelas</i>):
2	<i>positif</i> = <i>negatif</i> =0
3	for <i>i</i> in range(0,len(<i>kelas</i>)):
4	if <i>kelas</i> [<i>i</i>]==1:
5	<i>positif</i> = max(<i>positif</i> , <i>support_vector</i> [<i>i</i>])
6	else:
7	<i>negatif</i> = max(<i>negatif</i> , <i>support_vector</i> [<i>i</i>])
8	<i>xpositif</i> = <i>support_vector</i> .index(<i>positif</i>)
9	<i>xnegatif</i> = <i>support_vector</i> .index(<i>negatif</i>)
10	<i>jumlahpositif</i> = <i>jumlahnegatif</i> = 0
11	for <i>x</i> in range(0,len(<i>kelas</i>)):
12	<i>totalpositif</i> = <i>totalnegatif</i> =0
13	for <i>y</i> in range(0,len(<i>data</i> [<i>x</i>])):
14	<i>totalpositif</i> = <i>totalpositif</i> +(<i>data</i> [<i>x</i>][<i>y</i>]* <i>data</i> [<i>xpositif</i>][<i>y</i>])
15	<i>totalnegatif</i> = <i>totalnegatif</i> +(<i>data</i> [<i>x</i>][<i>y</i>]* <i>data</i> [<i>xnegatif</i>][<i>y</i>])
16	<i>kernelpositif</i> = (<i>totalpositif</i>)
17	<i>kernelnegatif</i> = (<i>totalnegatif</i>)
18	<i>hasilpositif</i> = <i>support_vector</i> [<i>x</i>]* <i>kelas</i> [<i>x</i>]* <i>kernelpositif</i>
19	<i>hasilnegatif</i> = <i>support_vector</i> [<i>x</i>]* <i>kelas</i> [<i>x</i>]* <i>kernelnegatif</i>
20	<i>jumlahpositif</i> += <i>hasilpositif</i>

21	jumlahnegatif += hasilnegatif
22	bias = (-0.5)*(jumlahnegatif+jumlahpositif)
23	return bias

Kode Program 5.12 Implementasi Perhitungan *bias*

Penjelasan potongan Kode Program 5.12 mengenai proses implementasi perhitungan *bias* diuraikan sebagai berikut:

1. Baris ke 4 sampai ke 9 menunjukkan proses perhitungan untuk menentukan *support vector* positif yang memiliki nilai paling tinggi dari *support vector* positif lainnya dan *support vector* negatif yang memiliki nilai paling tinggi dari *support vector* negatif lainnya.
2. Baris ke 11 sampai ke 15 menunjukkan proses perhitungan matriks antara matriks *ensemble feature* dengan matriks fitur berdasar *support vector* positif dan negatif.
3. Setelah itu dilakukan perhitungan kernel *linear* pada baris ke 16 dan 17.
4. Kemudian dilakukan perhitungan antara nilai *support vector* dan kelas dari data latih serta hasil perhitungan kernelnya pada baris ke 18 dan 19.
5. Setelah itu seluruh hasil perkalian tersebut dijumlahkan pada baris ke 20 dan 21.
6. Baris ke 23 menunjukkan proses perhitungan *bias*.

5.6.5 Implementasi Perhitungan Identifikasi

Perhitungan identifikasi adalah proses terakhir dalam proses *Support Vector Machine*. Proses ini melakukan identifikasi terhadap data uji yang diinputkan dan dilakukan identifikasi apakah data uji termasuk ujaran kebencian atau bukan ujaran kebencian. Implementasi perhitungan identifikasi ditunjukkan pada Kode Program 5.13.

Algoritme 13 : identifikasi	
1	def identifikasi(kelas,data,transpos,alpha,bias):
2	data_tes = data[len(kelas):
3	tes_hasil = []
4	hasil_identifikasi = []
5	for i in range(0,len(data_tes)):
6	row = []
7	for j in range(0,len(kelas)):
8	jumlah = 0
9	for k in range(0,len(data_tes[i])):
10	jumlah = jumlah +(data_tes[i][k]*transpos[k][j])

11	kernel = jumlah
12	hasil = alpha[j]*kelas[j]*kernel
13	row.append(hasil)
14	total = (sum(row))+bias
15	tes_hasil.append(total)
16	if total > 0.0:
17	kls = 'ujaran kebencian'
18	else :
19	kls = 'bukan ujaran kebencian'
20	hasil_identifikasi.append(kls)
21	return tes_hasil,hasil_identifikasi

Kode Program 5.13 Implementasi perhitungan identifikasi

Penjelasan potongan Kode Program 5.13 mengenai proses implementasi perhitungan identifikasi diuraikan sebagai berikut:

1. Baris ke 2 menunjukkan inisialisasi untuk matriks *ensemble feature* dari seluruh data uji.
2. Baris ke 8 sampai baris ke 14 menunjukkan proses perhitungan *kernel* dan perhitungan $\text{sign}(h(x))$ untuk seluruh data uji dan menyimpan hasilnya pada baris ke 15.
3. Baris ke 16 sampai baris ke 19 menunjukkan proses penentuan kelas data uji berdasarkan hasil perhitungan $\text{sign}(h(x))$. Jika nilai $\text{sign}(h(x))$ lebih dari 0 maka data uji masuk ke dalam kelas ujaran kebencian. Sedangkan jika nilai $\text{sign}(h(x))$ kurang dari 0 maka akan masuk ke dalam kelas bukan ujaran kebencian..

BAB 6 PENGUJIAN

Pada bab ini akan dijelaskan mengenai hasil dari proses pengujian terhadap sistem yang telah selesai dibuat dan proses analisis hasil dari pengujian yang dilakukan. Terdapat beberapa jenis pengujian yang dilakukan terhadap sistem yaitu pengujian pengaruh parameter *Support Vector Machine*, dan pengujian pengaruh jenis jenis *ensemble feature*, sedangkan untuk contoh hasil proses pengujian dapat dilihat pada Gambar 6.1.

```
Jumlah Kelas Ujaran Kebencian yang Teridentifikasi Benar adalah : 15
Jumlah Kelas Bukan Ujaran Kebencian yang Teridentifikasi Benar adalah : 1
Jumlah Kelas Ujaran Kebencian yang Teridentifikasi Salah adalah : 0
Jumlah Kelas Bukan Ujaran Kebencian yang Teridentifikasi Salah adalah : 14
Akurasi dari aplikasi adalah : 0.533333333333
Precision dari aplikasi adalah : 0.51724137931
Recall dari aplikasi adalah : 1.0
F-Measure dari aplikasi adalah : 0.681818181818
```

Gambar 6.1 Contoh Hasil Pengujian

6.1 Pengujian Pengaruh Parameter *Support Vector Machine*

Pengujian pengaruh parameter *Support Vector Machine* dilakukan untuk mendapatkan nilai-nilai parameter *Support Vector Machine* yang optimal untuk dipergunakan dalam proses identifikasi ujaran kebencian menggunakan metode *Support Vector Machine* yang kemudian akan menghasilkan akurasi sistem yang tinggi sehingga diketahui pengaruh dari parameter *Support Vector Machine* terhadap akurasi dari sistem. Parameter *Support Vector Machine* yang akan diujikan adalah nilai λ , nilai γ , dan nilai ϵ untuk perhitungan *sequential training SVM*. Untuk proses pengujian pengaruh parameter *Support Vector Machine* dijelaskan pada skenario pengujian dan analisis hasil pengujian.

6.1.1 Skenario Pengujian Pengaruh Parameter *Support Vector Machine*

Pada proses pengujian ini, parameter *Support Vector Machine* yang diujikan adalah nilai λ , nilai γ , dan nilai ϵ untuk perhitungan *sequential training SVM*. Pengujian ini menggunakan 150 data dengan menggunakan untuk melakukan validasi terhadap hasil parameter, dan jenis *Ensemble Feature* yang digunakan adalah gabungan dari fitur *Facebook*, *Tekstual*, dan *Lexicon*.

Pada pengujian nilai λ , nilai γ yang digunakan adalah 0,0001, serta nilai ϵ yang digunakan adalah 0,0001. Hasil dari pengujian nilai λ yang telah dilaksanakan dapat dilihat pada Tabel 6.1.

Tabel 6.1 Hasil Pengujian Nilai λ

	Nilai Lambda(λ)					Rata-rata
	0,5	0,7	0,9	1,1	1,3	
<i>Accuracy</i>	66 %	66 %	66 %	66 %	66 %	66 %
<i>Precision</i>	66 %	66 %	66 %	66 %	66 %	66 %
<i>Recall</i>	66 %	66 %	66 %	66 %	66 %	66 %
<i>F-Measure</i>	66 %	66 %	66 %	66 %	66 %	66 %

Sementara untuk pengujian nilai *learning rate*(γ), digunakan nilai *lambda*(λ) sebesar 0,5, dan nilai *epsilon*(ϵ) sebesar 0,0001. Hasil dari pengujian nilai *learning rate*(γ) ditunjukkan pada Tabel 6.2.

Tabel 6.2 Hasil Pengujian Nilai *learning rate*(γ)

	Nilai Learning Rate(γ)					Rata-Rata
	0,0001	0,0005	0,001	0,005	0,01	
<i>Accuracy</i>	66 %	66 %	66 %	50 %	50 %	60 %
<i>Precision</i>	66 %	65 %	63 %	50 %	50 %	59 %
<i>Recall</i>	66 %	73 %	80 %	6 %	6 %	46 %
<i>F-Measure</i>	66 %	69 %	70 %	11 %	11 %	45 %

Dan untuk pengujian nilai *epsilon*(ϵ), digunakan parameter nilai *lambda*(λ) sebesar 0,5, dan nilai *learning rate*(γ) sebesar 0,001. Hasil dari pengujian nilai *epsilon*(ϵ) ditunjukkan pada Tabel 6.3.

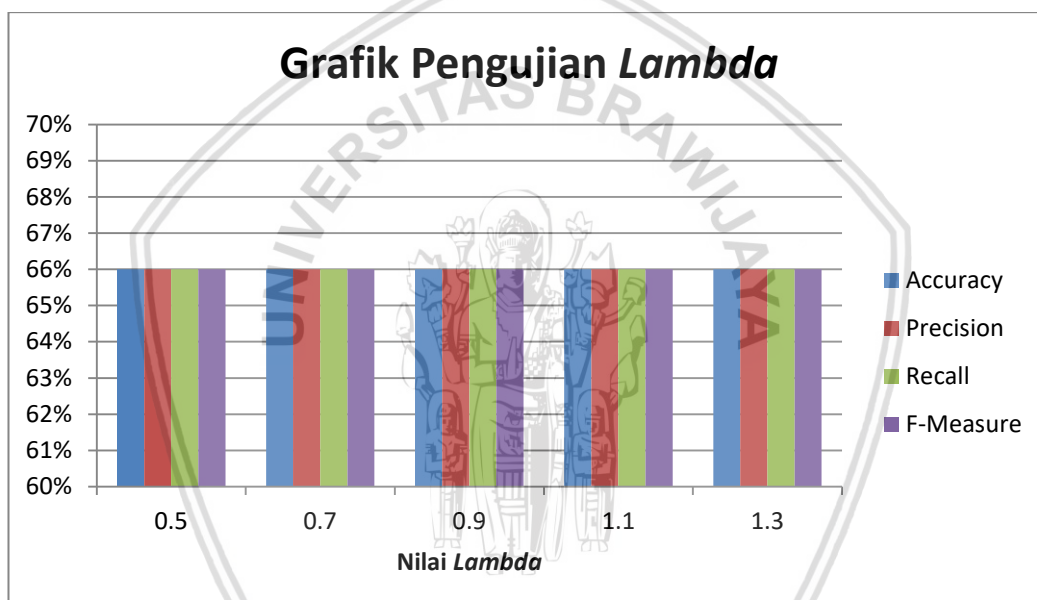
Tabel 6.3 Hasil Pengujian Nilai *epsilon*(ϵ)

	Nilai Epsilon(ϵ)					Rata-rata
	0,00005	0,0001	0,0005	0,001	0,005	
<i>Accuracy</i>	66 %	66 %	66 %	66 %	66 %	66 %
<i>Precision</i>	63 %	63 %	63 %	63 %	63 %	63 %
<i>Recall</i>	80 %	80 %	80 %	80 %	80 %	80 %
<i>F-Measure</i>	70 %	70 %	70 %	70 %	70 %	70 %

6.1.2 Analisis Hasil Pengujian Parameter *Support Vector Machine*

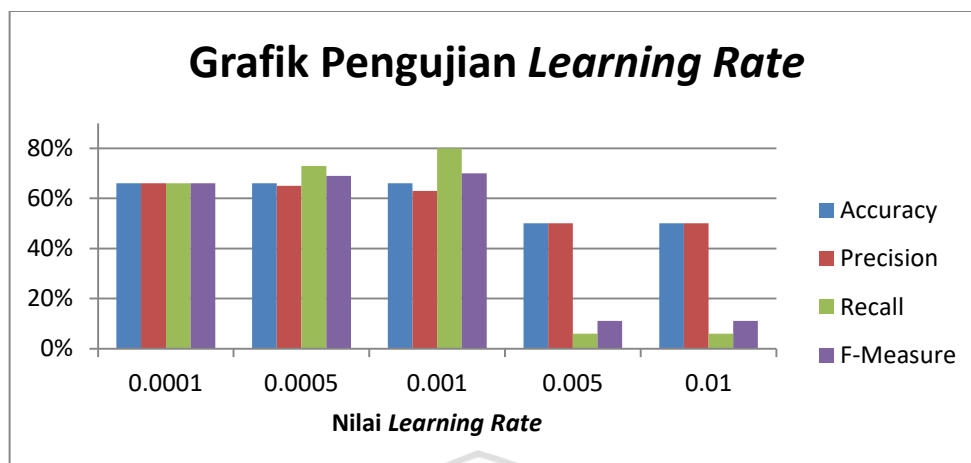
Berdasar hasil dari proses pengujian parameter *Support Vector Machine* yang dilaksanakan sebelumnya, yaitu pengaruh nilai λ , nilai γ , dan nilai ϵ untuk perhitungan *sequential training SVM* memberikan hasil akurasi yang bermacam-macam.

Hasil pengujian nilai λ pada Tabel 6.1 menunjukkan hasil akurasi yang sama berdasarkan nilai λ yang diinputkan. Diperoleh hasil rata-rata akurasi terbaik yaitu 66 %. Tidak terlihat adanya perbedaan pada hasil pengujian tiap nilai λ yang berbeda pada nilai akurasi, *precision*, dan *recall* pada data. Maka dari hasil pengujian nilai λ dapat dikatakan bahwa perubahan nilai λ tidak memiliki dampak yang signifikan terhadap perubahan pada nilai akurasi, *precision*, dan *recall* pada data. Grafik hasil pengujian pengaruh λ ditunjukkan pada Gambar 6.2.



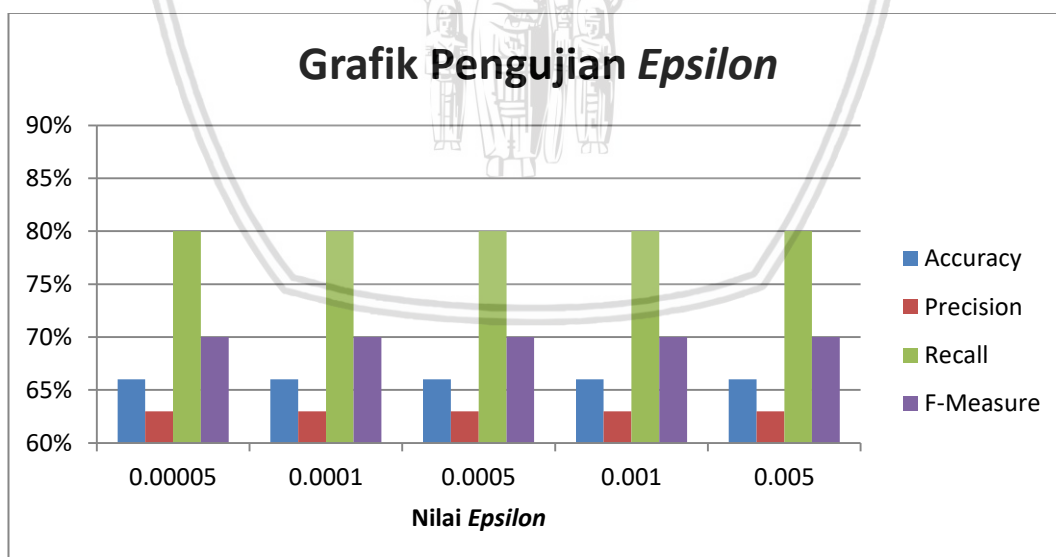
Gambar 6.2 Hasil pengujian pengaruh λ

Sedangkan pada hasil pengujian nilai γ pada Tabel 6.2 hasil rata-rata akurasi dan *f-measure* terbaik diperoleh saat nilai $\gamma < 0,005$. Pada saat $\gamma > 0,001$ terjadi penurunan akurasi saat γ 0,005, dan 0,01 dengan nilai akurasi berturut-turut adalah 50% dan 50%. Dari hasil pengujian diperoleh akurasi terbaik didapatkan jika nilai $\gamma < 0,005$, tetapi dalam praktiknya penerapan nilai γ yang terlalu rendah akan berimbas pada jumlah iterasi yang dilakukan oleh sistem dan akan berpengaruh pada jumlah waktu yang diperlukan untuk setiap proses identifikasi terhadap ujaran kebencian. Grafik hasil pengujian pengaruh γ ditunjukkan pada Gambar 6.3.



Gambar 6.3 Hasil pengujian pengaruh *learning rate*

Dan pada hasil pengujian nilai *epsilon*(ϵ) pada Tabel 6.3 diperoleh akurasi yang sama untuk setiap nilai *epsilon* yang berbeda-beda. Hal ini menunjukkan bahwa perbedaan nilai *epsilon* memiliki pengaruh yang sedikit terhadap nilai akurasi, *precision*, dan *recall* pada data. Tetapi besaran nilai *epsilon* berpengaruh pada jumlah iterasi yang dilakukan aplikasi untuk mencapai konvergensi pada data yang digunakan sebagai data latih. Hal ini secara tidak langsung juga berpengaruh pada waktu proses aplikasi karena semakin banyak iterasi yang dilakukan akan memakan waktu yang tidak sedikit juga. Grafik hasil pengujian *epsilon*(ϵ) ditunjukkan pada Gambar 6.4.



Gambar 6.4 Hasil pengujian pengaruh *epsilon*

6.2 Pengujian Pengaruh Jenis *Ensemble Feature*

Pengujian pengaruh jenis *Ensemble Feature* dilakukan untuk melihat jenis *Ensemble Feature* mana yang akan menghasilkan akurasi terbaik dengan menggunakan *Support Vector Machine*. Jenis *Ensemble Feature* yang akan diuji ada 4 jenis yaitu *Ensemble Feature* yang terbentuk dari fitur *Facebook*, *Tekstual*, dan *Lexicon*, fitur *Facebook*, dan *Lexicon*, fitur *Tekstual*, dan *Lexicon*, dan fitur *Facebook*, dan *Tekstual*. Untuk proses pengujian jenis *Ensemble Feature* dijelaskan lebih lanjut pada bagian skenario pengujian dan analisis hasil pengujian

6.2.1 Skenario Pengujian Pengaruh *Ensemble Feature*

Pada proses pengujian ini, Jenis *Ensemble Feature* dan fitur individual yang akan diuji masing-masing ada 4 jenis dan 3 jenis yaitu *Ensemble Feature* yang terbentuk dari fitur *Facebook*, *Tekstual*, dan *Lexicon*. Pengujian ini menggunakan 150 data dengan menggunakan 120 data sebagai data latih dan 30 data sebagai data uji. Nilai dari parameter *Support Vector Machine* yang digunakan dalam pengujian ini adalah nilai optimal dari tiap parameter *Support Vector Machine* hasil pengujian sebelumnya yaitu nilai λ sebesar 0,5, nilai γ sebesar 0,001, dan nilai ϵ sebesar 0,0001 untuk perhitungan *sequential training SVM*. Hasil dari pengujian pengaruh jenis *Ensemble Feature* dapat dilihat pada Tabel 6.4.

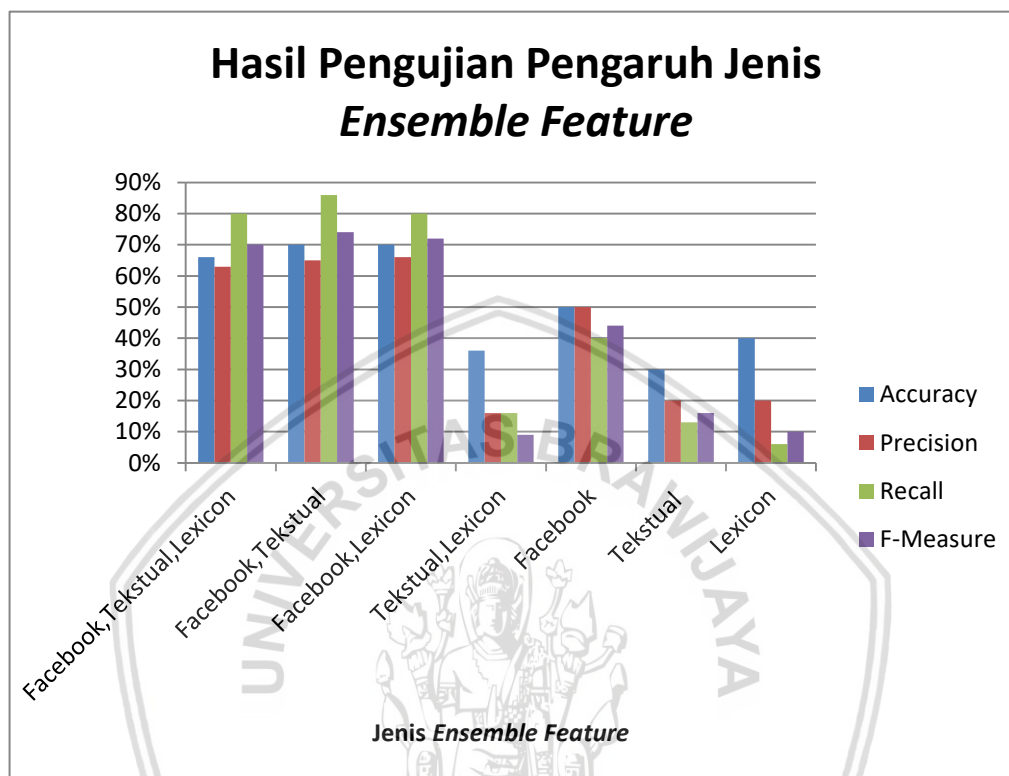
Tabel 6.4 Hasil Pengujian Pengaruh Jenis *Ensemble Feature*

	Jenis <i>Ensemble</i>							Rata-rata
	<i>Facebook</i> , <i>Tekstual</i> , <i>Lexicon</i>	<i>Facebook</i> , <i>Tekstual</i>	<i>Facebook</i> , <i>Lexicon</i>	<i>Tekstual</i> , <i>Lexicon</i>	<i>Facebook</i>	<i>Tekstual</i>	<i>Lexicon</i>	
<i>Accuracy</i>	66%	70%	70%	36%	50%	30%	40%	52 %
<i>Precision</i>	63%	65%	66%	16%	50%	20%	20%	43 %
<i>Recall</i>	80%	86%	80%	16%	40%	13%	6%	46 %
<i>F-Measure</i>	70%	74%	72%	9%	44%	16%	10%	42 %

6.2.2 Analisis Hasil Pengujian Pengaruh Jenis *Ensemble Feature*

Berdasarkan hasil pengujian pengaruh jenis *Ensemble Feature* pada subbab sebelumnya, diperoleh hasil berupa akurasi yang beragam dari jenis *Ensemble Feature*. Dari hasil pengujian didapatkan akurasi tertinggi dengan menggunakan jenis *Ensemble Feature* yang terdiri dari fitur *Facebook*, dan *Tekstual*, serta fitur *Facebook*, dan *Lexicon*. Dengan menggunakan jenis *Ensemble Feature* lain dapat terlihat terjadi penurunan pada akurasi untuk tiap jenis *Ensemble Feature*, hal ini terjadi karena penggunaan 2 kategori fitur (yaitu fitur *Tekstual* dan *Lexicon*) tidak cukup akurat untuk memisahkan 2 kelas data karena

untuk sebagian data memiliki nilai yang perbedaannya cukup dekat sehingga akan menyebabkan kurang akuratnya SVM dalam menentukan *hyperplane* dari 2 kelas data yang berbeda karena memiliki nilai fitur yang hampir sama. Grafik hasil pengujian jenis *Ensemble Feature* ditunjukkan pada Gambar 6.4.



Gambar 6.5 Hasil pengujian pengaruh jenis *Ensemble Feature*

BAB 7 PENUTUP

Pada bab terakhir ini akan diberikan uraian untuk kesimpulan dari penelitian yang dilakukan dan saran yang dapat dipergunakan untuk proses pengembangan penelitian berikutnya.

7.1 Kesimpulan

Berdasarkan hasil pengujian yang telah dilakukan sebelumnya, maka dapat diambil sebuah kesimpulan mengenai identifikasi ujaran kebencian pada *Facebook* dengan metode *Ensemble Feature* dan *Support Vector Machine* yaitu:

1. Untuk parameter optimal bagi metode *support vector machine* didapatkan angka untuk masing-masing parameter yaitu untuk parameter $\lambda = 0,5$, untuk parameter *learning rate* = 0,001 dan nilai parameter *epsilon* = 0,0001, nilai ini di dapatkan melalui pengujian dengan berbagai nilai dari masing-masing parameter *support vector machine* untuk mendapatkan hasil akurasi terbaik dari penggunaan parameter yang optimal untuk proses identifikasi *support vector machine*.
2. Untuk penelitian identifikasi ujaran kebencian pada *Facebook* dengan metode *Support Vector Machine* dan *Ensemble Feature*, diperoleh nilai akurasi identifikasi ujaran kebencian dengan nilai akurasi, *precision*, dan *recall* terbaik masing-masing dengan nilai 70%, 65%, dan 86% dengan menggunakan *ensemble feature* yang terdiri dari kategori fitur *Facebook*, dan tekstual. Jika dibandingkan dengan nilai akurasi, *precision*, dan *recall* dari kategori fitur individual yang relatif lebih rendah yaitu dengan nilai akurasi, *precision*, dan *recall* untuk fitur *Facebook* secara berturut-turut adalah, 50%, 50%, dan 40%. Sedangkan untuk fitur tekstual adalah 30%, 20%, dan 13%. Serta fitur *lexicon* dengan nilai 40%, 20%, dan 6%.

7.2 Saran

Berdasarkan hasil dari penelitian yang dilakukan terdapat beberapa kekurangan yang dapat dilakukan perbaikan dan atau pengembangan untuk proses penelitian berikutnya. Beberapa saran untuk proses penelitian berikutnya yaitu:

1. Menambahkan jumlah dari data latih untuk memperoleh hasil akurasi yang lebih baik
2. Menambahkan data latih dengan nilai fitur yang berbeda-beda untuk tiap data agar hasil akurasi lebih baik. Dari penelitian ini diperoleh hasil akurasi yang kurang baik pada penggunaan kategori fitur tekstual, dan *lexicon* hal ini dikarenakan nilai kedua kategori fitur tidak berbeda terlalu jauh sehingga mempengaruhi akurasi dari sistem

DAFTAR PUSTAKA

- Aula, P., 2010. *Social Media, Reputation Risk and Ambient Publicity Management. Strategy & Leadership Vol.: 38 Iss.: 6 (2010) 43-49*, [e-journal]. Academia.edu. Tersedia di:<<http://www.academia.edu/download/35256169/Aula-Strategy-and-Leadership.pdf>>[Diakses 20 Agustus 2017]
- Arifin dan Purnama, K.E., 2012. *Classification of Emotion in Indonesian Texts Using K-NN Method*. Tersedia Melalui: Universitas Dian Nuswantoro Repository<<http://eprints.dinus.ac.id/14094/1/classification-of-emotion-using-KNN.pdf>>[Diakses 20 Agustus 2017]
- Asian, J., 2007. *Effective Techniques for Indonesian Text Retrieval*. S2. RMIT University. Tersedia di<researchbank.rmit.edu.au>[Diakses 20 Agustus 2017]
- Cortes, C. dan Vapnik, V., 1995. *Support Vector Networks. Machine Learning Vol.:20 Iss.: 3(1995) 273-297*[e-journal]. Tersedia melalui: Springer Link<<https://link.springer.com>>[Diakses 20 Agustus 2017]
- Huang, X. dan Zhang, L., 2013. *An SVM Ensemble Approach Combining Spectral, Structural, and Semantic Features for the Classification of High-Resolution Remotely Sensed Imagery. IEEE Transactions on Geoscience and Remote Sensing Vol. 51 No. 1, (Januari 2013)*[e-journal]. Tersedia melalui: IEEE Explore<<http://ieeexplore.ieee.org/document/6239588/>>[Diakses 11 Agustus 2017]
- Kent, A Berry, M Luehrs, F U & Perry, J W., 1955. *Machine literature searching. VIII: operational criteria for designing information retrieval systems. American Documentation. Iss.: VI, Vol.: 2, 93-101*.
- Liu, B., Hu, M., and Cheng, J., 2005. *Opinion Observer: Analyzing and Comparing Opinions on the Web. Proceedings of the 14th International World Wide Web Conference, (2005) 342-351*[e-journal]. Tersedia melalui: ACM Digital Library<<https://dl.acm.org/citation.cfm?id=1060797>>[Diakses 06 Agustus 2017]
- Metz, Charles E. , 1978. *Basic principles of ROC analysis. Seminars in nuclear medicine Vol. 8. Iss. 4 283-298*.
- Quan-Haase, A. dan Young, A.L., 2010. *Uses and Gratifications of Social Media: A Comparison of Facebook and Instant Messaging. Bulletin of Science, Technology & Society Vol.: 30 Iss.: 5, (Oktober 2010) 350-361*[e-journal]. Tersedia melalui: ResearchGate<<https://www.researchgate.net/publication/211906609>>[Diakses 18 Agustus 2017]

- Saeys, Y., Abeel, T. & Peer, Y.V., 2008. *Robust Feature Selection Using Ensemble Feature Selection Techniques*. *Machine Learning and Knowledge Discovery in Databases(2008)*[e-journal]. Tersedia melalui: Springer Link<<https://link.springer.com>>[Diakses 11 Agustus 2017]
- Serbes, G., Sakar, C.O., Kahya, Y.P., Aydin, N., 2011. *Feature Extraction using Time-Frequency/Scale Analysis and Ensemble of Feature Sets for Crackle Detection*. *33rd Annual International Conference of the IEEE EMBS(2011)*[e-journal]. Tersedia melalui: ResearchGate<<https://www.researchgate.net/publication/221756920>>[Diakses 11 September 2017]
- Siddiqua, U.A., Ahsan, T. & Chy, A.N., 2016. *Combining a Rule-based Classifier with Ensemble of Feature Sets and Machine Learning Technique Sentiment Analysis on Microblog*. *19th International Conference on Computer and Information Technology(2016)*[e-journal]. Tersedia melalui: ResearchGate<<https://www.researchgate.net/publication/313953662>>[Diakses 08 September 2017]
- Tsybmal, A., Puuronen, S. & Patterson, D.W., 2003. *Ensemble Feature selection with the simple Bayesian classification*. *Information Fusion Vol.: 4 Iss.: 2, (2003) 87-100* [e-journal]. Tersedia melalui: ScienceDirect<<https://www.sciencedirect.com/science/article/pii/S1566253503000046>>[Diakses 11 Agustus 2017]
- Vijayakumar, S & Wu, S 1999, Sequential support vector classifiers and regression. in *Proc. International Conference on Soft Computing (SOCO'99), Genoa, Italy*.
- Warner, W. dan Hirschberg, J., 2012. *Detecting Hate Speech on the World Wide Web*. *Proceedings of the 2012 Workshop on Language in Social Media, (Juni, 2012) 19-26*[e-journal]. Tersedia melalui: Association for Computational Linguistics (ACL)<<http://www.aclweb.org/anthology/W12-2103>>[Diakses 06 Agustus 2017]
- Wahid, D. H., & Azhari, S. N., 2016. *Peringkasan Sentimen Esktraktif di Twitter Menggunakan Hybrid TF-IDF dan Cosine Similarity*. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems) Vol.: 10 No.: 2,(2016) 207-218*. Tersedia melalui: IJCCS (Indonesian Journal of Computing and Cybernetics Systems)<<https://journal.ugm.ac.id/ijccs/article/view/16625>>[Diakses 06 Agustus 2017]
- Zia, T., Akram, M.S., Nawaz, M.S., Shahzad, B., Abdullatif, A.M., Mustafa, R.U., Lali, M.I., 2017. *Identfication of Hatred Speeches on Twitter*. *International Journal of Advances in Electronics and Computer Science Vol.: 4 Iss.: 1, (Januari, 2017) 27-32*[e-journal]. Tersedia melalui: ResearchGate<<https://www.researchgate.net/publication/311169981>>[Diakses 18 September 2017]

LAMPIRAN

LAMPIRAN 1 Tabel Kalimat Data Latih

No.	Kalimat
1	AHMAD DHANI MINTA KAUM CEKON LEGOWO MENERIMA KEKALAHAN SUPER TELAK AHOK DARI ANIES-SANDI.
2	ASAL TAU AJA YA MANA KELOMPOK YANG BENAR-BENAR MEMBELA UMAT ISLAM DAN MANA YANG HANYA MEMIKIRKAN POLITIK DAN MERAIH KEKUASAAN.
3	Semuanya difitnah Penebar SARA dengan ujaran kebencian
4	2. Metro TV gencar membentuk opini bahwa yang terjadi adalah kekerasan dan bukan genosida, kekerasan berarti menyalahkan korban dan pembantai sekaligus.
5	Mereka menjadi TUYUL raksasa yang sangat sadis & ganas.
6	Hidup yang tidak bermanfaat itu adalah yang dihabiskan hanya untuk jadi "badut" di media sosial tanpa pernah membantu sesama.
7	MEDIA BAJINGAN & PENJAHAT SOSMED
8	Bagi Cecunguk Koh Napi, Menebar Fitnah Setiap Hari Hukumnya Adalah WAJIB!
9	Efek kekalahan telak Ahok dari Gubernur Muslim Anies Baswedan pada Pilkada DKI 2017 lalu nampaknya membuat Gubernur Kalimantan Barat, Cornelis yang beragama Katolik meradang.
10	Wakwaw! Saat Karangan Bunga kiriman Cukong Aseng DII untuk Zhong-Djarot dibakar-bakar oleh massa buruh di depan Balaikota DKI siang tadi para Cecunguk Zhong hanya bisa nyinyir dan ngedumel di medsos, tanpa ada satu pun yang berani mendatangi apalagi menggagalkan aksi pembakaran oleh buruh.
11	Pasca KEKALAHAN SUPER TELAK sang majikan dari Anies-Sandi di Pilkada DKI 2017, serangan dan badai fitnah kembali masif digencarkan oleh para Cecunguk Terdakwa terutama melalui media-media online, terhadap Imam Besar FPI Habib Muhammad Rizieq Shihab.
12	Berikut ini adalah MEDIA MEDIA SAMPAH yang besok akan MENGGORENG ISU dan FITNAH MURAHAN tentang HABIB RIZIEQ untuk MENUTUPI KASUS SIDANG AHOK, diantaranya:
13	Penundaan Sidang Penista Agama dengan agenda pembacaan Rencana Tuntutan oleh JPU dengan alasan belum siap karena belum selesai diketik

	adalah TRAGEDI PENEGAKAN HUKUM.
14	TAPI PERTARUNGAN ANTARA: PEMBELA ISLAM vs PENISTA ISLAM
15	Ahoker Gagal Move On Denny Siregar berusaha mengadu domba umat Kristiani vs umat Islam.
16	POLITIK BALAS DENDAM 1. Semua pejabat atau pegawai yang pro Ulama yang jatuhkan Ahok harus dipecat.
17	SINETRON PENEGAKAN HUKUM! Sesuai dugaan banyak kalangan terutama usai dikirimnya surat Kapolda Metro Jaya kepada Majelis Hakim minta sidang Terdakwa Zhong harus ditunda, akhirnya hal ini pun benar-benar terealisasi.
18	Ini adalah seorang Mahasiswi di Jakarta dari total sekitar tujuh belas orang korban lainnya yang telah berhasil dikristenkan oleh para missionaris gereja.
19	Mungkinkah ini yang menyebabkan Panglima TNI ngotot banget mengadakan nobar film G30SPKI, sampai2 beliau bilang "Emang gue pikirin" terhadap orang2 yang tak setuju?
20	Dari acara #ILCPKI tadi malam, saya makin sadar bahwa salah satu kebiasaan orang PKI adalah BERBOHONG.
21	Jika membahas PKI dituduh sebagai upaya untuk menyerang pemerintah, maka tanpa sadar para IQ 200 Sekolam ini berpendapat bahwa pemerintah RI itu PKI.
22	Mungkin yang dimaksud oleh Bapak Joko Widodo adalah Gerakan 30 September 2017, bukan 1965.
23	Demikian Aturan tentang Hina Menghina di Era Ikan Tongkol.
24	Tahukah Anda bahwa bapak presiden kita tercinta Joko Widodo beberapa hari lalu berkunjung ke Singapura?
25	Masyarakat Jawa Barat itu religius, dekat dengan nilai-nilai Islam.
26	Ada orang yang marah pada saya karena menurutnya saya memplesetkan namanya dari AIDID menjadi AIDIT.
27	Myanmar Berani Membasmi Muslim Rohingya karena Pemerintah Indonesia Tidak Berwibawa Itulah salah satu fakta menarik yang disampaikan oleh Ketua Umum FPI KH Ahmad Shabri Lubis saat melakukan orasi #SaveMuslimRohingya bersama Ormas Bang Japar, di depan Kedubes Myanmar tanggal 8 September 2017.
28	Ahok yang bla... bla.. bla kalian demo berjilid-jilid. First Travel yang bla... bla... justru tidak pernah kalian demo. Kenapa? ujar para bani kotak-

	kotak.
29	Salah satu bentuk PENISTAAN AGAMA adalah ketika kita sibuk mengomentari bahkan nyinyirin ajaran agama orang lain.
30	Artis cerai, yang dibahas cadarnya.
31	3. Hal Biasa Tokoh Masyarakat Menemui Habib Rizieq, Karena Kalau Yang Mereka Temui Si DeSi Atau Abu Janda Bisa APESSS Dan KALAH SUPER TELAK Lagi Seperti Ahok-Djarot Kemarin.
32	Padahal sebelumnya sang pasangan incumbent yang didukung total oleh penguasa sudah dengan sombongnya berkoar-koar "AHOK-DJAROT PASTI MENANG MUDAH SATU PUTARAN!".
33	Dilantik Pun BELUM, Tapi Badai Serangan Dan Fitnah Terhadap Anies-Sandi Sudah Sangat Masif Dilancarkan Oleh Media Cekon (CEcunguk KOH Napi) Dan Buzzer-Buzzernya.
34	Cecunguk Owe Makin Blo'on Aja.
35	Jika PKI sudah tidak ada lagi di Indonesia, lantas kenapa kok banyak yang blingsatan ketika kita mau memutar film G30SPKI ???
36	Mungkin kalian sudah demikian takutnya kehilangan kekuasaan ya? Kok segala sesuatu dihubung-hubungkan dengan politik?.
37	Tapi saya ingin melihat masalah ini secara sederhana saja. Jika ada orang yang demikian gampangnya menuduh orang lain sebagai pecandu PCC dan FLAKKA, maka mungkin dia...
38	Mereka adalah SARACEN yang sebenarnya.
39	Berarti yang bikin fitnah itu kagak ngerti aturan agama.
40	Mereka pasti sedang menggoreng isu untuk menjatuhkan Presiden Jokowi.
41	Gerombolan CEBONG SARACEN BIADAB makin sadis dan licik.
42	Kalau Jokowi Sang Junjungan Ente itu sudah menganggap bahwa Jonru gak perlu diurus, jadi ngapain kalian masih sibuk aja ngurusin Jonru?.
43	Kubu merekalah yang paling rajin menebar kebencian, menebar hoax, sehingga merekalah yang paling layak untuk jadi saracen.
44	Hoax dan kebencian dari kubu mereka dipelihara, sementara dari pihak kita... ya gitu deh.
45	JANGAN DIKIRA KAMI BODOH, begitu mudahnya tertipu oleh trik licik kalian yang sebenarnya sangat murahan dan kampungan itu.

46	Lomba Panjat Pinang Diciptakan oleh Penjajah Belanda untuk Menertawakan Pribumi?
47	Ada yang nanya gini ke saya, "Kalo ada perusahaan travel yang nilep dana haji dan umroh, itu termasuk penistaan agama atau tidak? Jika ada pejabat yang korupsi dana haji, itu termasuk penistaan agama atau tidak?".
48	Mulai sekarang, istilah "9 Naga" sudah tidak berlaku.
49	Fakta Sejarah: Inilah Perampok NKRI yang Sebenarnya
50	Hari ini, ternyata tambah satu orang lagi. Namanya Sugriwan Soedarmo.
51	Sementara itu, laporan Bank Dunia sempat menyatakan bahwa Indonesia menghadapi dua masalah yang serius dan sulit ditemukan jalan keluarnya.
52	Lalu saya ngutang yang banyak ke luar negeri, sehingga negara lain bebas mendikte Indonesia Saya Pancasila, NKRI Harga Mati.
53	Di dalam pesawat, persis di depan saya, duduk tiga orang Tionghoa; dua perempuan (yang fotonya saya candid ini, ketika pesawat sudah landing), dan satu pria yang tak terlihat pad foto.
54	Pakde ini Islam atau bukan sih? Kalau Islam, pasti paham bahwa menyalahgunakan biaya haji itu hukumnya haram.
55	Ketika tadi saya menulis artikel tentang "HTI vs PKI", banyak banget orang yang marah-marah dan bertanya, "Mana PKI-nya? Tunjukkan buktinya.
56	Breaking News: Setyo Novanto ditetapkan sebagai tersangka korupsi eKTP.
57	Masih ingatkah Anda ketika dulu Jokowi BARU SAJA terpilih jadi presiden dan belum dilantik? Pendiri Facebook Mark Zuckerberg langsung datang ke Indonesia untuk menemuinya.
58	Nyapres lagi di tahun 2019, timses Jokowi sangat sadar bahwa mereka akan sulit menang, sebab saat ini kebangkitan Islam di Indonesia makin terang-benderang.
59	Tersiar kabar bahwa Boni Hargens (Komisaris Kantor Berita Antara) tiba-tiba sakau ketika siaran live di TV One.
60	Dear Jokowers, Anda bangga terhadap banyaknya infrastruktur yang dibangun oleh rezim Jokowi.
61	KONYOL plus GOBLOG adalah ketika para haters sibuk nuduh saya ini itu, tapi tak pernah berani melaporkan ke polisi.
62	Memang para haters ini otaknya udah konslet semua.

63	Makanya mereka bisa seenaknya menghina kita, sambil menuduh kita tukang hina.
64	Jangan-jangan ente emang antek-antek penjajah.
65	Ayolah, kita jangan niru kubu mereka yang cacat logika.
66	Tentang sebuah ormas radikal dan intoleran yang hobi membubarkan pengajian.
67	Jakarta, Hanter 寰 • Kalangan pengamat menyebut popularitas Presiden Joko Widodo (Jokowi) belakangan semakin tergerus dan dinilai berada pada level terendah. g.
68	Kepolisian berkata bahwa si Viktor yang intoleran rasis, dan radikal itu tidak bisa diadili karena dia punya hak imunitas.
69	Lebih kelewatan lagi karena perbuatan intolerannya itu justru didukung dan dibela oleh teman-teman satu gengnya.
70	Jika engkau makin panik, makin bingung, makin kelabakan, gak tahu lagi gimana cara bayar utang yang jumlahnya luar biasa, yang sebentar lagi jatuh tempo, dan negara sudah tidak ada lagi dana untuk membayarnya, sehingga dana haji pun mau diembat juga....
71	Islam dizalimi, dituduh teroris, dituduh anti NKRI, dituduh anti Pancasila, ormasnya dibubarkan, ulamanya dikriminalisasi, katanya beragama tak boleh serius, agama harus pisah dari politik...
72	Termasuk jika mereka semua DIJUAL ke China untuk membayar utang Indonesia, mereka pasti rela-rela aja kok.
73	Ternyata yang benar adalah: Saya Cinta China, NKRI Mau Mati Karena Kebanyakan Ngutang pada China.
74	Argumen ente sungguh cacat logika plus dangkal pula.
75	Dan jika si pemilik dana tidak ridho, maka silahkan menunggu ADZAB YANG SANGAT PEDIH DARI ALLAH.
76	Kopi Amerika yang (Ternyata) Buatan Indonesia
77	Ketika polisi berkata bahwa Ahok tetap di Mako Brimob karena di Cipinang banyak musuhnya, maka SUNGGUH BANYAK SEKALI argumen skakmat yang bisa kita berikan:
78	TAKBIR KELILING DILARANG, NAMUN PAWAI LILIN MALAM HARI, DAN PESTA TAHUN BARU DIIZINKAN.
79	Beberapa tahun belakangan ini, entah kenapa tiba-tiba isu "kebangkitan PKI" muncul lagi.

80	Dari mana @Metro_TV bisa mengarang berita saya mau lantik FPI di Bitung?.
81	Tapi ustadz yang mengingatkan bahaya PKI kok malah ditahan.
82	Ada orang dari kubu sebelah yang mendesak pemerintah untuk membubarkan Muslim Cyber Army (MCA).
83	Nih, saya nemu lagi satu posting yang membuktikan bahwa Para Ahokers Kalap Gagal Move On dan Jongos Asing Aseng ini cuma bisa nyusahin orang banyak.
84	Wahai Penguasa, Tidakkah Kalian Belajar dari Kasus LHI?
85	Ahok dipindah ke Mako Brimob dengan alasan situasi tidak aman, disebabkan banyaknya pendukung Ahok yang demo sampai pagi di depan lapas.
86	Terima kasih untuk para pendemo di rutan cipinang yang rela bertahan sampai pagi tanpa takut diusir oleh polisi. Karena jasa Anda yang rela demo sampai pagi, Ahok pun dipindah ke Mako Brimob.
87	Negara Federal Republik Papua Barat (NFRPB) Surati Presiden Joko Widodo
88	SeBAGUS dan seMAHAL apapun karangan bunga yang engkau persembahkan, akhirnya hanya akan menjadi SAMPAH yang mengotori kota, bikin repot banyak orang.
89	Ahok yang katanya bersih dan tegas terhadap koruptor itu, berani gak ya... teriak "IBU MALING !!!" terhadap si GADIS AHOK bernama Miryam S Haryani ini, yang baru saja diserahkan oleh Bareskrim ke KPK?
90	Justru banyak pemikiran HTI dan perilaku para kadernya yang tidak saya sukai.
91	Dua sikap yang saling kontradiktif, bukan? Karena mereka emang MENTAL PELACUR.
92	Dulu ketika Ahok jadi tersangka, mereka ngotot menghendaki agar doi tetap jadi gubernur.
93	Emang masih ada yang mau milih dia? Kalau ada, berarti orangnya pasti sudah cacat otak.
94	Dan yang bikin saya EMOSI LUAR BIASA: Para jahannam dari kubu sebelah tiba-tiba memfitnah istri pak Hermawan, membunuh karakternya dengan membuat isu yang sangat buruk.
95	Bayangkan apa yang ada di pikiran Anda ketika berhadapan dengan seorang presiden dari negara lain yang gayanya seperti itu.

96	Kedunguan Jokowi makin permanen.
97	Kata mereka, "Jonru kejang-kejang karena idolanya Erdogan ketemu sama Jokowi."
98	Namun sebelum memindahkan ibukota RI ke Kalimantan, coba itu si Ahok dipindah dulu ke Cipinang.
99	Tapi Bapak Presiden Joko Widodo ternyata lebih suka bertemu dengan Afi yang terbukti PLAGIATOR.
100	Tajam ke umat Islam, tumpul ke bani kotak-kotak.
101	Padahal dengan adanya pembatalan naik banding, Ahok seharusnya secara resmi menjadi narapidana dan dijebloskan ke penjara.
102	Maka kita pun semakin yakin bahwa rezim ini sedang BALAS DENDAM kepada para ulama yang menggerakkan Aksi Bela Islam.
103	HEHEHEHE.. MEREKA EMANG JAGO MEMELINTIR FAKTA DAN MEM-FRAMING INFORMASI UNTUK MENYERANG KITA.
104	Lantas ketika kecewa karena jagoannya masuk penjara, dengan santainya dia nyerocos, "Aku sudah tak punya semangat untuk berbakti pada negeri ini. Kelak anakku akan kusuruh berbakti ke negara lain saja. Di negara ini sudah tidak ada harapan."
105	Tapi kenapa Om kagak nongol-nongol juga? Apa Om lagi makan ongol-ongol? Cucok deh Om:-)
106	Mereka yang selama ini lebih sering tidak terlihat di depan publik, tiba-tiba muncul semuanya di TPS.
107	Bunuh diri dong namanya, kalo ada partai sarang koruptor yang berani mendukung pemimpin yang tegas dan berani membasmi korupsi.
108	CARA-CARA LICIK selalu mereka gunakan untuk MENGEMIS-NGEMIS suara dari umat Islam
109	Mereka pun cari muka ke NU dengan cara seperti terlihat pada gambar.
110	Menjual Banjir untuk Kepentingan Politik Itu Cetek Banget, Sob!
111	Kabar Gembira untuk muslim pendukung Ahok.
112	Namun ternyata dia berhasil menunjukkan kebodohnya sendiri.
113	Baju Kotak-Kotak yang Kebal Hukum
114	Karena itulah, kita harus berjuang, membela kebenaran, membasmi kemungkaran.
115	Haters berkicau, "Ngakunya anti china, tapi HP-nya buatan china. Situ

	waras?"
116	Maka para pemujanya pun kelabakan plus kehilangan muka.
117	Kalau mau memisahkan agama dari politik, seharusnya itu dilakukan sebelum masa kampanye.
118	Sedang rame-ramenya hashtag #boikotinuldaratista, karena artis goyang ngebor yang satu ini telah menghina ulama.
119	Kayaknya dulu para haters pernah ngomong gini ke saya, "Eh Jonru. Ente kok cuma mengkritik Jokowi dan Ahok? Kenapa ente gak pernah mengkritik pemimpin lain, seperti Ridwan Kamil misalnya?"
120	Masih ingatkah Anda, ketika dulu mereka membully, menghina dan mencaci-maki AHY, namun tiba-tiba memuji-mujinya setinggi langit, dengan harapan agar suara pendukung AHY disalurkan ke kubu mereka?

LAMPIRAN 2 Tabel Kalimat Data Uji

No.	Kalimat
1	Namun di balik nama besar gerakan sosial yang dia dirikan, dan telah banyak berperan dalam memajukan Indonesia, ternyata dia sama saja dengan para ahokers lainnya yang terkena penyakit CINTA BUTA.
2	BANI SERbet memang perlu dibubarkan, karena merekalah yang radikal, intoleran dan berjiwa teroris.
3	Namun yang bikin saya SANGAT AMAT HERAN BANGET SEKALI: Kenapa kok pendukung Jokowi rajin banget membully dan menyalahkan SBY?
4	Sementara si penista agama yang sudah jadi terdakwa, justru masih bebas berkeliaran.
5	Sebab presidennya udah ganti, tapi mereka masih aja sibuk mengkritik, menyerang, membully dan mendemo mantan presiden.
6	NB: Memang belakangan ini, banyak banget orang kafir yang sok tahu dan mengomentari ajaran Islam berdasarkan sudut pandang mereka sendiri. Sebenarnya ini bagian dari PENISTAAN AGAMA.
7	Sepertinya emang ada Bani Kotak-Kotak yang sedang butuh popularitas, sehingga dia terus memancing-mancing dan memprovokasi saya untuk meladeni tingkahnya.
8	BANI KOTAK-KOTAK EMANG BENAR-BENAR TAK BEROTAK.
9	Bani Kotak-Kotak Lagi-Lagi Terbukti Tak Berotak.

10	Jika si penista agama tidak dihukum juga, maka penistaan demi penistaan seperti ini akan terus terjadi.
11	Jika Anda masih dengan SOK IMUTNYA mendukung PELACURAN, coba BAYANGKAN aja deh: Seandainya ibu anda jadi pelacur Seandainya anak perempuan anda jadi pelacur Seandainya saudara perempuan anda jadi pelacur Seandainya bibi atau tante anda jadi pelacur Seandainya istri anda jadi pelacur.
12	Dulu ketika kampanye, sowan ke pesantren, pakai sorban, umroh super ekspres, foto sedang shalat disebar ke mana-mana, istri pun dipakaikan jilbab Setelah terpilih, ulama pun dikriminalisasi, islam makin ditekan, makin dibungkam.
13	Ada Komunitas Anti hoax anti hasut anti fitnah, anggotanya adalah para tukang hoax dan tukang fitnah.
14	Para Ahokers emang hobi BERDUSTA.
15	Kebayang gak sih? Jokowi terlihat sangat lelah seperti ini, hanya gara-gara ulah seorang penista agama bernama Ahok?
16	Saran saya, coba sunat dulu.
17	Cie cie.....:-)
18	Selama ini para ahokers melakukan manipulasi terhadap istilah "rahmatan lil 'alamin".
19	Mengapa Bani Kotak-Kotak terus berusaha membangun pencitraan terhadap Ahok, padahal junjungan mereka sudah kalah Pilgub?
20	Saya pernah dituduh kafir, dituduh syiah, dituduh nyamar jadi orang Islam, dituduh tukang fitnah, dituduh provokator, dan sebagainya.
21	Pencitraan Lewat Karangan Bunga
22	KEKALAHAN yang paling menyakitkan (bahkan bisa bikin gila) adalah ketika kita sangat yakin jagoan kita tak terkalahkan.
23	Banyak pendukung Ahok yang berharap agar junjungannya jadi gubernur Bali, jadi mendagri, jadi ketua KPK dan seterusnya.
24	SEJUJURNYA, ada satu hal yang membuat saya geregetan pasca kemenangan Anies-Sandi.
25	Siapa bekingnya? Tentu saja sembilan cacing, seperti yang dulu pernah disampaikan oleh Kwik Kian Gie.
26	Para Ahokers langsung sibuk menagih janji-janji kampanye Anies Sandi. Padahal belum dilantik.

27	Mereka mengusulkan pada Presiden Jokowi agar Ahok diangkat jadi menteri.
28	Bahkan hingga hari ini, mereka masih seperti itu.
29	Aneh! Beri Kesaksian dalam Sidang, Ketua MUI Malah Di Ancam Akan Dilaporkan oleh Ahok
30	GAYA FITNAH DAN ADU DOMBA ALA PKI SEMAKIN NYATA DI NEGERI INI

LAMPIRAN 3 Tabel Fitur Facebook dan Kelas Data Latih

No	<i>Like React</i>	<i>Angry React</i>	Jumlah <i>Share</i>	Jumlah <i>Comment</i>	Kelas
1	2,000	0	185	80	Negatif
2	2,000	12	984	185	Negatif
3	822	3	235	50	Negatif
4	1,200	4	256	38	Negatif
5	6,800	697	2,990	1,172	Negatif
6	211	27	50	41	Negatif
7	3,200	266	1,181	500	Negatif
8	4,900	409	2,098	198	Negatif
9	3,000	166	942	90	Negatif
10	681	224	686	210	Negatif
11	3,300	4	2,023	974	Negatif
12	7,300	60	2,402	538	Negatif
13	1,200	112	505	74	Negatif
14	1,100	45	325	26	Negatif
15	2,200	114	2,473	296	Negatif
16	544	121	239	51	Positif
17	319	7	57	12	Positif
18	1,400	9	358	35	Positif
19	3,100	28	684	285	Positif
20	72	0	260	12	Positif
21	149	0	510	14	Positif
22	37,000	73	10,366	3,429	Positif
23	16,000	160	4,447	4,898	Positif

24	13,000	10	1,689	576	Positif
25	12,000	836	3,045	921	Positif
26	20,000	140	3,549	2,701	Positif
27	12,000	13	2,284	1,126	Positif
28	16,000	8	1,639	4,311	Positif
29	13,000	6	683	2,191	Positif
30	6,100	11	856	341	Positif
31	677	23	165	49	Negatif
32	1,700	4	327	163	Negatif
33	4,400	0	720	350	Negatif
34	2,100	27	939	143	Negatif
35	19,000	82	4,091	1441	Negatif
36	26,000	4	11,123	1,330	Negatif
37	16,000	1	1,745	731	Negatif
38	3,900	82	1,266	1,082	Negatif
39	22,000	4	2,363	1,356	Negatif
40	9,300	1	569	836	Negatif
41	5,100	12	605	559	Negatif
42	17,000	10	3,147	1,556	Negatif
43	7,200	3	307	518	Negatif
44	17,000	10	3,459	2,646	Negatif
45	5,900	40	878	635	Negatif
46	9,100	3	394	719	Positif
47	4,800	1	70	287	Positif
48	32,000	23	12,751	2,163	Positif
49	10,000	109	5,473	2,315	Positif
50	3,700	1	90	320	Positif
51	9,300	10	947	1,013	Positif
52	20,000	31	5,124	3,060	Positif
53	9,700	48	3,535	933	Positif
54	18,000	25	4,448	2,007	Positif
55	24,000	35	10,088	4,166	Positif

56	17,000	159	5,940	2,270	Positif
57	21,000	22	5,151	2,214	Positif
58	17,000	12	3,345	1,821	Positif
59	14,000	11	2,554	2,388	Positif
60	16,000	30	6,654	1,686	Positif
61	9,500	18	1,042	791	Negatif
62	9,100	3	1,455	886	Negatif
63	5,300	2	555	416	Negatif
64	27,000	133	10,881	1,730	Negatif
65	12,000	75	1,792	975	Negatif
66	6,200	1.4	1,924	709	Negatif
67	6,400	2	660	504	Negatif
68	13,000	8	2,289	1,535	Negatif
69	12,000	89	3,906	932	Negatif
70	11,000	3	1,609	1,265	Negatif
71	12,000	19	3,051	1,714	Negatif
72	14,000	369	5,826	1,912	Negatif
73	7,400	387	1,307	603	Negatif
74	16,000	113	5,379	1,661	Negatif
75	16,000	821	15,084	1,896	Negatif
76	22,000	79	6,360	4,978	Positif
77	13,000	75	3,585	2,018	Positif
78	22,000	20	4,399	3,582	Positif
79	15,000	48	5,400	1,731	Positif
80	13,000	2,900	9,344	2,784	Positif
81	10,000	109	3,091	1,166	Positif
82	26,000	288	12,838	2,675	Positif
83	14,000	470	4,167	927	Positif
84	6,900	17	1,349	716	Positif
85	11,000	63	2,336	1,552	Positif
86	14,000	352	7,684	905	Positif
87	30,000	146	14,679	6,017	Positif

88	14,000	232	6,215	1,353	Positif
89	22,000	216	7,857	1,675	Positif
90	17,000	17	5,384	1,711	Positif
91	22,000	22	3,300	2,058	Negatif
92	23,000	181	7,630	2,950	Negatif
93	14,000	583	6,041	3,846	Negatif
94	15,000	15	2,247	1,854	Negatif
95	10,000	4	1,721	1,437	Negatif
96	13,000	183	3,198	1,758	Negatif
97	11,000	13	1,966	1,195	Negatif
98	16,000	64	3,731	5,857	Negatif
99	17,000	12	4,311	1,431	Negatif
100	7,600	1	1,386	610	Negatif
101	21,000	49	3,527	1,450	Negatif
102	14,000	269	6,731	1,039	Negatif
103	12,000	668	3,800	1,114	Negatif
104	27,000	163	12,502	1,440	Negatif
105	8,700	1	1,607	499	Negatif
106	20,000	25	5,081	2,228	Positif
107	8,800	158	982	776	Positif
108	46,000	65	20,393	8,265	Positif
109	33,000	37	16,610	4,227	Positif
110	7,900	158	3,904	671	Positif
111	25,000	31	12,708	1,888	Positif
112	14,000	177	13,929	1,395	Positif
113	47,000	197	41,633	16,345	Positif
114	26,000	21	13,752	2,030	Positif
115	25,000	29	16,002	4,768	Positif
116	23,000	37	24,774	8,168	Positif
117	16,000	128	8,004	1,301	Positif
118	9,300	31	1,608	289	Positif
119	8,400	1	1,523	1,006	Positif

120	17,000	3	4,762	1,326	Positif
-----	--------	---	-------	-------	---------

LAMPIRAN 4 Tabel Fitur Facebook dan Kelas Data Uji

No	<i>Like React</i>	<i>Angry React</i>	Jumlah <i>Share</i>	Jumlah <i>Comment</i>	Kelas
1	19,000	30	6,973	1,661	Negatif
2	6,100	65	2,913	718	Negatif
3	11,000	50	5,690	938	Negatif
4	26,000	484	17,512	2,223	Negatif
5	31,000	11	5,375	3,269	Negatif
6	12,000	156	5,322	416	Negatif
7	17,000	59	5,160	2,724	Negatif
8	8,800	5	2,081	1,191	Negatif
9	1,800	0	41	211	Negatif
10	3,100	5	326	276	Negatif
11	17,000	28	3,975	919	Negatif
12	8,700	113	3,053	627	Negatif
13	29,000	31	13,455	2,341	Negatif
14	32,000	1,500	15,280	3,616	Negatif
15	2,600	5	89	381	Negatif
16	19,000	143	11,440	1,405	Positif
17	20,000	316	6,852	3,657	Positif
18	2,300	1	152	230	Positif
19	9,400	6	1,486	654	Positif
20	23,000	474	16,483	5,973	Positif
21	11,000	2	3,751	314	Positif
22	2,100	0	67	183	Positif
23	9,200	15	2,908	2,646	Positif
24	6,300	7	1,103	1,084	Positif
25	15,000	6	6,797	1,071	Positif
26	3,000	0	353	270	Positif
27	9,900	6	1,358	742	Positif
28	4,200	1	396	402	Positif

29	13,000	2,100	5,922	2,245	Positif
30	47,000	19	24,887	3,809	Positif



SURAT KESEDIAAN PAKAR

Saya yang bertanda tangan di bawah ini:

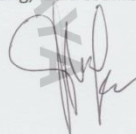
Nama : Dr. Eti Setiawati, M.Pd
Nomor Identitas : 19640413 199203 2 001
Jabatan : Dosen Program Studi Pendidikan Bahasa dan Sastra
Indonesia Bidang Linguistik Mikropragmatik

Dengan ini saya menyatakan bersedia sebagai pakar untuk menunjang data penelitian skripsi/tugas akhir mahasiswa yang berjudul "**Identifikasi Ujaran Kebencian pada Facebook dengan Metode Ensemble Features dan Support Vector Machine**" yang dilakukan oleh:

Nama : Aditya Kresna Bayu Arda Putra
NIM : 135150200111028

Demikian pernyataan yang saya buat untuk dapat dipergunakan sebagaimana mestinya.

Malang, 22 Desember 2017



Dr. Eti Setiawati, M.Pd
NIP: 19640413 199203 2 001

